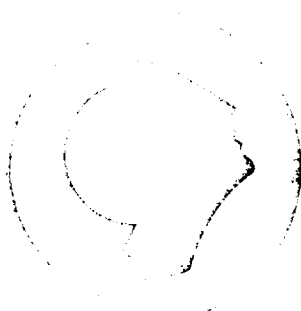


D. O. 123

NAS 8-3 69SS



**Users Manual for Program SSFREQ
Intermediate Mode Stability Curves
Developed for Use on a PC Computer**

(NASA-CR-184452) USERS MANUAL FOR
PROGRAM SSFREQ INTERMEDIATE MODE
STABILITY CURVES: DEVELOPED FOR USE
ON A PC COMPUTER Final Report
(Alabama Univ.) 85 p

N93-12675

Unclass

G3/61 0127408

1993-12675

Research Institute
The University of Alabama in Huntsville

87

**Users Manual for Program SSFREQ
Intermediate Mode Stability Curves
Developed for Use on a PC Computer**

Wilbur C. Armstrong

June 1992

Table of Contents

Section	Page
1.0 Introduction	1
2.0 Input Description	2
2.1 Description of ENG.RLN file	2
2.2 Description of LOX.RLN and FUEL.RLN files	2
2.3 Description of IMODE.RLN file	4
2.4 Description of IMODE.FRQ file	4
2.5 Description of IMODE.TAU file	4
3.0 Output Description	5
3.1 Output files (IMODE.OUT & SURF.ERR)	5
3.2 Graphs Available	6
4.0 Sample Run	6
4.1 Input Files for Sample Run	6
4.2 Walkthrough of Sample Run	9
4.3 Output for Sample Run	23
5.0 Flow Diagram	29
6.0 Variable Descriptions	30
7.0 Program Listing	46

1.0 Introduction

The piping in a liquid rocket can assume complex configurations due to multiple tanks, multiple engines, and structures that must be piped around. The capability to handle some of these complex configurations have been incorporated into the SSFREQ code. The capability to modify the input on line has been implemented.

The configurations allowed include multiple tanks, multiple engines, the splitting of a pipe into unequal segments going to different (or the same) engines. This program will handle the following type elements

- Straight pipes
- Bends
- Inline accumulators
- Tuned stub accumulators
- Helmholtz resonators
- Parallel resonators
- Pumps
- Split pipes
- Multiple tanks
- Multiple engines

2.0 Input Description

SSFREQ uses the following files: ENG.RLN, LOX.RLN, FUEL.RLN, IMODE.RLN, and optionally uses IMODE.FRQ and IMODE.TAU. All files are in free format, therefore each of the following records will give the same results.

Record 1: 1.000000E-01 6219.000000 2.670000 2.330E-03 -315.0000

Record 2: 0.1 6219.0 2.67 0.00233 -315.0

Record 3: 1.E-01
6219.0
2.67
2.33E-3
-315.0

The file assignments are given in the following table:

Unit	File Name	File Type	Description
9	ENG.RLN	Input	Engine data
10	LOX.RLN	Input	LOX tanks & lines data
11	FUEL.RLN	Input	Fuel tanks & lines data
12	IMODE.RLN	Input	Chamber data
13	SURF.ERR	Output	Convergence error information
14	IMODE.OUT	Output	n-tau values
15	(LOX)	Work	Temporary file with LOX data
16	(FUEL)	Work	Temporary file with fuel data
17	(RESULT)	Work	Temporary file for results
18	IMODE.TAU	Input	tau's to be used
19	IMODE.FRQ	Input	Frequencies to be used
20	(IMODE)	Work	Temporary file for chamber data

2.1 Description of file ENG.RLN

Card # 1
number of engines
Card # 2
total flow in engine (lbm/sec),
chamber pressure (lbf/ft²),
pressure drop across orifice (lbf/ft²)
Read card # 2 "number of engines" times

2.2 Description of files LOX.RLN or FUEL.RLN

Card # 1
title
Card # 2
number of tanks

type	name	PIPE1	PIPE2	PIPE3	PIPE4	PIPE5
0	bend	radius	angle	diameter	end len.	
1	straight	length	diameter			
2	inline	length	diameter			
3	tuned	length	diameter			
4	Helmholtz	length	diameter	volume		
5	parallel	length	diameter	volume		
6	pump	length	diameter	$dp/d\dot{m}$	L	C
7	manifold	volume	bulk mod.			

Dimensions:

radius, length, diameter, end length	- ft
angle	- deg
volume	- ft ³
$dp/d\dot{m}$ (non-dimensionalized by \bar{m}/\bar{p}_C)	- non-dimensional
L	- sec
C	- sec
bulk modulus	- lbf/ft ²

2.3 Description of file IMODE.RLN

Card # 1

title

Card # 2

number of x stations

Card # 3

x - location (ft)

pressure (lbf/ft²)

temperature (°R)

Read card # 3 "number of x stations" times

Card # 4

invariant time lag (sec)

mixture ratio interaction index

damping part of frequency

Card # 5

chamber diameter (ft)

throat diameter (ft)

length of combustion chamber (ft)

Card # 6

ratio of specific heats

gas constant ((ft/sec)²/°R)

maximum overpressure (lbf/ft²)

mixture ratio

PRECEDING PAGE BLANK NOT FILMED

Card # 7
 dc^*/dr (ft/sec)
 dh/dr ((ft/sec)²)
 mass of liquid per unit chamber volume (lbm/ft³)
 axial component of liquid velocity (ft/sec)
 Read card # 2-7 "number of engines" times

2.4 Description of file IMODE.FRQ

Card # 1
 number of frequencies
 Card # 2
 frequency (rad/sec or Hertz)
 Read card # 2 "number of frequencies" times

2.5 Description of file IMODE.TAU

Card # 1
 number of taus (sensitive time lags)
 Card # 2
 tau (sec)
 Read card # 2 "number of taus" times

3.0 Output Description

3.1 Output Files

Output from the program is a file (IMODE.OUT) which may be printed and various graphs under the control of the user. The print file contains the following:

CAREA=
 TAREA=
 ASTAR=
 CSTARD=
 RHOBAR=
 UBAR=
 RHOBAR=
 UBAR=

Title, time, and date

Engine No.

DIMENSIONAL VARIABLES

NVAL=
 XBAR=
 UBAR=
 DTAU =
 TDIAM =
 NR =
 XLC =
 CDIAM =
 GAMMA =

RGAS =	P00 =	MBAR =
RBAR =	DCSDR =	DHLDLDR =
RHOLO =	ULO =	PCHMB =
TCHMB =		

NON-DIMENSIONAL VARIABLES

NVAL=		
XBAR=		
UBAR=		
DTAU=	NR=	RBAR=
MBAR=	GAMMA=	P00=
DHLDLDR=	CSTAR=	DCSDR=
RHOLO=	ULO=	
RFA=		RFC=

the above for each engine

FREQUENCY =

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
----------	---------	---	---------	---------

one entry for each engine and each tau requested
the above for each frequency requested.

Also, if a split pipe is analyzed, a file (SURF.ERR) is created if any point fails to converge within the specified number of iterations.
This file contains:

Title, time, and date

jw =		after	iterations has error of	% in	line
	I=	J=	G =	GOLD =	

3.2 Graphs Available

The graphs available are

1. Plot of the n-tau curve for a given frequency and engine upon request.
2. Plot of the n-tau curves for each engine. All frequencies for an engine are plotted on one graph.

4.0 Sample Run

The sample run consists of two lox tanks and four engines, two of the engines and lines going to them are identical. The total mass flow from each tank is the same, however the line from the first tank is split with half the mass flow going to engine # 2 and the other half split into two identical engines # 1.

4.1 Input for Sample Run

Sample ENG.RLN file:

```
3
853.5      4.502040E+05      1.610532E+06
1707.0     4.502040E+05      1.610532E+06
3414.0     4.502040E+05      1.610532E+06
```

Sample LOX.RLN file:

```
Sample Run
2
1.956300E+04  2928.0  1.185883E+07  71.4
1.956300E+04  2928.0  1.185883E+07  71.4
2
1 0
13 2
1 15.0      1.416      0.0      0.0      0.0
0 35.0      45.0      1.416      0.0      0.0
1 30.0      1.416      0.0      0.0      0.0
0 3.5       135.0     1.416      0.0      0.0
1 15.0      1.416      0.0      0.0      0.0
1 20.641    1.416      0.0      0.0      0.0
1 20.558    1.416      0.0      0.0      0.0
1 20.558    1.416      0.0      0.0      0.0
1 8.541     1.416      0.0      0.0      0.0
1 6.383     1.416      0.0      0.0      0.0
0 4.25      90.0      1.416      0.0      0.0
1 9.33      1.416      0.0      0.0      0.0
0 3.33      80.0      1.416      0.0      0.0
5 1 1
1 3.53      0.708      0.0      0.0      0.0
1 12.2      0.708      0.0      0.0      0.0
0 1.28      35.0      0.708      0.0      0.0
1 12.2      0.708      0.0      0.0      0.0
7 13.5     1.183346E+07  0.0      0.0      0.0
5 1 2
1 3.53      1.00126 0.0      0.0      0.0
1 12.2      1.00126 0.0      0.0      0.0
0 1.28      35.0      1.00126 0.0      0.0
1 12.2      1.00126 0.0      0.0      0.0
7 13.5     1.183346E+07  0.0      0.0      0.0
2 3
18 0
1 15.0      1.416      0.0      0.0      0.0
0 35.0      45.0      1.416      0.0      0.0
1 30.0      1.416      0.0      0.0      0.0
0 3.5       135.0     1.416      0.0      0.0
1 15.0      1.416      0.0      0.0      0.0
1 20.641    1.416      0.0      0.0      0.0
1 20.558    1.416      0.0      0.0      0.0
1 20.558    1.416      0.0      0.0      0.0
```

1	8.541	1.416	0.0	0.0	0.0
1	6.383	1.416	0.0	0.0	0.0
0	4.25	90.0	1.416	0.0	0.0
1	9.33	1.416	0.0	0.0	0.0
0	3.33	80.0	1.416	0.0	0.0
1	3.53	1.416	0.0	0.0	0.0
1	12.2	1.416	0.0	0.0	0.0
0	1.28	35.0	1.416	0.0	0.0
1	12.2	1.416	0.0	0.0	0.0
7	13.5	1.183346E+07	0.0	0.0	0.0

Sample FUEL.RLN file:

```

FUEL Split: 3-1,1-0
2
4.055000E+03  486.0  1.185883E+07  72.13
4.055000E+03  486.0  1.185883E+07  72.13
2
1 0
8 2
1 17.97  1.04154 0.0  0.0  0.0
0 1.2785  75.0  1.04154 0.0  0.0
1 8.138  1.04154 0.0  0.0  0.0
0 1.2785 -75.0  1.04154 0.0  0.0
1 32.51  1.04154 0.0  0.0  0.0
0 1.2785  75.0  1.04154 0.0  0.0
1 8.65  1.04154 0.0  0.0  0.0
0 1.2785 -75.0  1.04154 0.0  0.0
2 1 1
1 6.2  0.52077 0.0  0.0  0.0
7 4.5  1.183346E+07 0.0  0.0  0.0
2 1 2
1 6.2  0.73648 0.0  0.0  0.0
7 4.5  1.183346E+07 0.0  0.0  0.0
2 3
10 0
1 17.97  1.04154 0.0  0.0  0.0
0 1.2785  75.0  1.04154 0.0  0.0
1 8.138  1.04154 0.0  0.0  0.0
0 1.2785 -75.0  1.04154 0.0  0.0
1 32.51  1.04154 0.0  0.0  0.0
0 1.2785  75.0  1.04154 0.0  0.0
1 8.65  1.04154 0.0  0.0  0.0
0 1.2785 -75.0  1.04154 0.0  0.0
1 6.2  1.04154 0.0  0.0  0.0
7 4.5  1.183346E+07 0.0  0.0  0.0

```

Sample IMODE.RLN file:

```

Check Case for SSFREQ
2
0.000000  450204.00  4000.000
4.00000  450204.00  4000.000

```

0.000697	0.01	0.000000	
3.214000	2.232000	4.000000	
1.200000	1716.000	142500.0	2.670000
-315.0000	0.010000	0.440000	1965.000
2			
0.000000	450204.00	4000.000	
4.00000	450204.00	4000.000	
0.000697	0.01	0.000000	
3.214000	2.232000	4.000000	
1.200000	1716.000	142500.0	2.670000
-315.0000	0.010000	0.440000	1965.000
2			
0.000000	450204.00	4000.000	
4.00000	450204.00	4000.000	
0.000697	0.01	0.000000	
3.214000	2.232000	4.000000	
1.200000	1716.000	142500.0	2.670000
-315.0000	0.010000	0.440000	1965.000

Sample IMODE.FRQ file:

3
110 113 116

Sample IMODE.TAU file:

11
0.0005 0.0007 0.0009 0.0011 0.0013 0.0015
0.0017 0.0019 0.0021 0.0023 0.0025

4.2 Walkthrough of Sample Run

Welcome to SSFREQ - an Intermediate Mode Program

To send a plot to the printer

The computer MUST be in GRAPHICS mode

Hit PrScn to send the current plot to the printer

If you want frequency in rad/sec, hit enter.
 If you want it in Hertz, enter "H". h
 Is the engine data on file ENG.RLN? (Y/N) y
 Is fuel line data in a file? (Y/N) y
 Is the file name FUEL.RLN? (Y/N) y
 Max. no. of iterations is set at 20
 Do you wish to change it? n
 Is lox line data in a file? (Y/N) y

Is the file name LOX.RLN? (Y/N) y
Max. no. of iterations is set at 20
Do you wish to change it? n

Are you are using IMODE.RLN for input data? y

Welcome to SSFREQ

Intermediate Mode Rocket Stability Aide

There are three types of input, rocket parameters,
Oxidizer feed parameters, and fuel feed parameters,
Each may be read from files or from the keyboard

File Name	Input
IMODE.RLN or NAME read in	Rocket Parameters
LOX.RLN	Oxidizer Parameters
FUEL.RLN	Fuel Parameters

If keyboard entry, you will be prompted for values

Is your rocket input on file? Y OR N y
Does the file need to be rewound? Y OR N n

Sample Run 12:35PM 12-12-91

Engine No. 1

DIMENSIONAL VARIABLES

NVAL=	2		
XBAR=	0.00000E+00	4.00000E+00	
UBAR=	4.98522E+01	4.98522E+01	
DTAU =	6.97000E-04	NR = 1.00000E-02	CDIAM = 3.21400E+00
TDIAM =	2.23200E+00	XLC = 4.00000E+00	GAMMA = 1.20000E+00
RGAS =	1.71600E+03	P00 = 1.42500E+05	MBAR = 8.53500E+02
RBAR =	2.67000E+00	DCSDR = -3.15000E+02	DHLDR = 1.00000E-02
RHOLO =	4.40000E-01	ULO = 1.96500E+03	PCHMB = 4.50204E+05
TCHMB =	4.00000E+03		

NON-DIMENSIONAL VARIABLES

NVAL=	2		
XBAR=	0.00000E+00	1.00000E+00	
UBAR=	1.73702E-02	1.73702E-02	
DTAU=	5.00094E-01	NR= 1.00000E-02	RBAR= 2.67000E+00
MBAR=	6.94809E-02	GAMMA= 1.20000E+00	P00= 3.16523E-01
DHLDR=	1.00000E-02	CSTAR= 2.31372E+01	DCSDR= -1.09757E-01
RHOLO=	2.08505E-01	ULO= 6.84673E-01	
RFA=	1.44752E-03	0.00000E+00	RFC= 0.00000E+00 0.00000E+00

Hit ENTER to continue

Sample Run

12:35PM 12-12-91

Engine No. 2

DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 4.00000E+00
UBAR= 9.97045E+01 9.97045E+01
DTAU = 6.97000E-04 NR = 1.00000E-02 CDIAM = 3.21400E+00
TDIAM = 2.23200E+00 XLC = 4.00000E+00 GAMMA = 1.20000E+00
RGAS = 1.71600E+03 P00 = 1.42500E+05 MBAR = 1.70700E+03
RBAR = 2.67000E+00 DCSDR = -3.15000E+02 DHLDR = 1.00000E-02
RHOLO = 4.40000E-01 ULO = 1.96500E+03 PCHMB = 4.50204E+05
TCHMB = 4.00000E+03

NON-DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 1.00000E+00
UBAR= 3.47404E-02 3.47404E-02
DTAU= 5.00094E-01 NR= 1.00000E-02 RBAR= 2.67000E+00
MBAR= 1.38962E-01 GAMMA= 1.20000E+00 P00= 3.16523E-01
DHLDR= 1.00000E-02 CSTAR= 1.15686E+01 DCSDR= -1.09757E-01
RHOLO= 2.08505E-01 ULO= 6.84673E-01
RFA= 2.89504E-03 0.00000E+00 RFC= 0.00000E+00 0.00000E+00
Hit ENTER to continue

Sample Run

12:35PM 12-12-91

Engine No. 3

DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 4.00000E+00
UBAR= 1.99409E+02 1.99409E+02
DTAU = 6.97000E-04 NR = 1.00000E-02 CDIAM = 3.21400E+00
TDIAM = 2.23200E+00 XLC = 4.00000E+00 GAMMA = 1.20000E+00
RGAS = 1.71600E+03 P00 = 1.42500E+05 MBAR = 3.41400E+03
RBAR = 2.67000E+00 DCSDR = -3.15000E+02 DHLDR = 1.00000E-02
RHOLO = 4.40000E-01 ULO = 1.96500E+03 PCHMB = 4.50204E+05
TCHMB = 4.00000E+03

NON-DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 1.00000E+00
UBAR= 6.94809E-02 6.94809E-02
DTAU= 5.00094E-01 NR= 1.00000E-02 RBAR= 2.67000E+00
MBAR= 2.77924E-01 GAMMA= 1.20000E+00 P00= 3.16523E-01
DHLDR= 1.00000E-02 CSTAR= 5.78429E+00 DCSDR= -1.09757E-01
RHOLO= 2.08505E-01 ULO= 6.84673E-01
RFA= 5.79008E-03 0.00000E+00 RFC= 0.00000E+00 0.00000E+00
Hit ENTER to continue

Specify how frequency will be input -

Enter R for a range of values
Enter F for values in a file

Enter K (end with -999) to enter values from keyboard

f

Is the frequency on IMODE.FRQ?

Enter Y or N y

Specify how tau will be input -

Enter R for a range of values

Enter F for values in a file

Enter K to enter values from keyboard

f

Is tau on IMODE.TAU?

Enter Y or N y

FREQUENCY = 1.10000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.14621E+01	-3.11896E-07	-4.67076E-01
2	5.00000E-04	1.14841E+01	-3.03067E-07	-4.60113E-01
3	5.00000E-04	1.14920E+01	3.88765E-07	-4.57809E-01
1	7.00000E-04	6.55967E+00	8.66792E-07	-3.17231E-01
2	7.00000E-04	6.48823E+00	-2.23960E-06	-3.05747E-01
3	7.00000E-04	6.46472E+00	-1.18901E-06	-3.01928E-01
1	9.00000E-04	4.62124E+00	2.49193E-07	-2.20846E-01
2	9.00000E-04	4.50544E+00	6.42385E-07	-2.08742E-01
3	9.00000E-04	4.46697E+00	-1.25355E-06	-2.04704E-01
1	1.10000E-03	3.73689E+00	-2.28634E-07	-1.44484E-01
2	1.10000E-03	3.59644E+00	-2.45890E-07	-1.35610E-01
3	1.10000E-03	3.54970E+00	-7.22797E-07	-1.32642E-01
1	1.30000E-03	3.35282E+00	1.55936E-07	-7.53268E-02
2	1.30000E-03	3.20046E+00	1.72578E-07	-7.29305E-02
3	1.30000E-03	3.14967E+00	-2.02976E-07	-7.21169E-02
1	1.50000E-03	3.28318E+00	-4.40167E-08	-7.92099E-03
2	1.50000E-03	3.13166E+00	5.27691E-08	-1.37797E-02
3	1.50000E-03	3.08109E+00	5.24031E-08	-1.57178E-02
1	1.70000E-03	3.49309E+00	-2.72676E-09	6.03049E-02
2	1.70000E-03	3.36009E+00	-3.73278E-09	4.66884E-02
3	1.70000E-03	3.31559E+00	-2.31758E-08	4.21583E-02
1	1.90000E-03	4.06690E+00	2.46680E-08	1.31826E-01
2	1.90000E-03	3.98268E+00	1.20805E-08	1.13762E-01
3	1.90000E-03	3.95433E+00	-1.35946E-08	1.07738E-01
1	2.10000E-03	5.30008E+00	-2.17681E-08	2.12487E-01
2	2.10000E-03	5.32637E+00	-6.04931E-09	1.96344E-01
3	2.10000E-03	5.33471E+00	-1.82410E-08	1.90944E-01
1	2.30000E-03	8.12702E+00	-2.26766E-08	3.19093E-01
2	2.30000E-03	8.41822E+00	-8.19192E-09	3.14610E-01
3	2.30000E-03	8.51465E+00	-2.97111E-08	3.13079E-01
1	2.50000E-03	1.64494E+01	-1.18208E-10	5.06763E-01
2	2.50000E-03	1.75388E+01	3.19730E-08	5.29430E-01
3	2.50000E-03	1.79009E+01	3.66275E-10	5.36935E-01

Do you wish to see n vs tau for this frequency? y

Specify which engine you wish to view

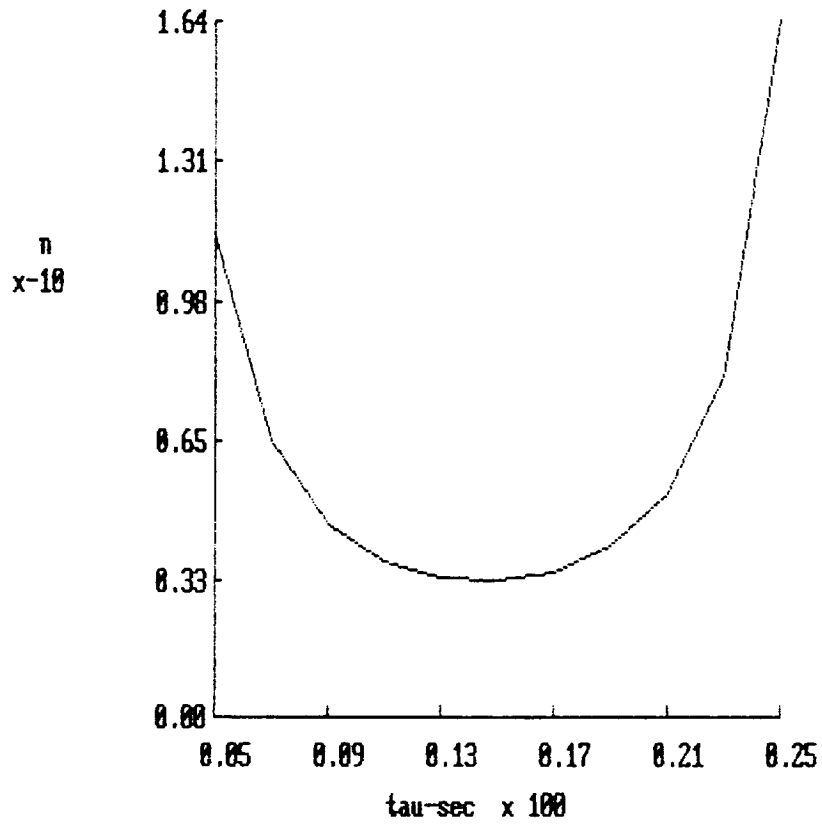
Enter 1 - 3 or 0 to continue 1

Sample Run

12:35PM 12-12-91

Engine No. 1

frequency = 110.000 Hertz



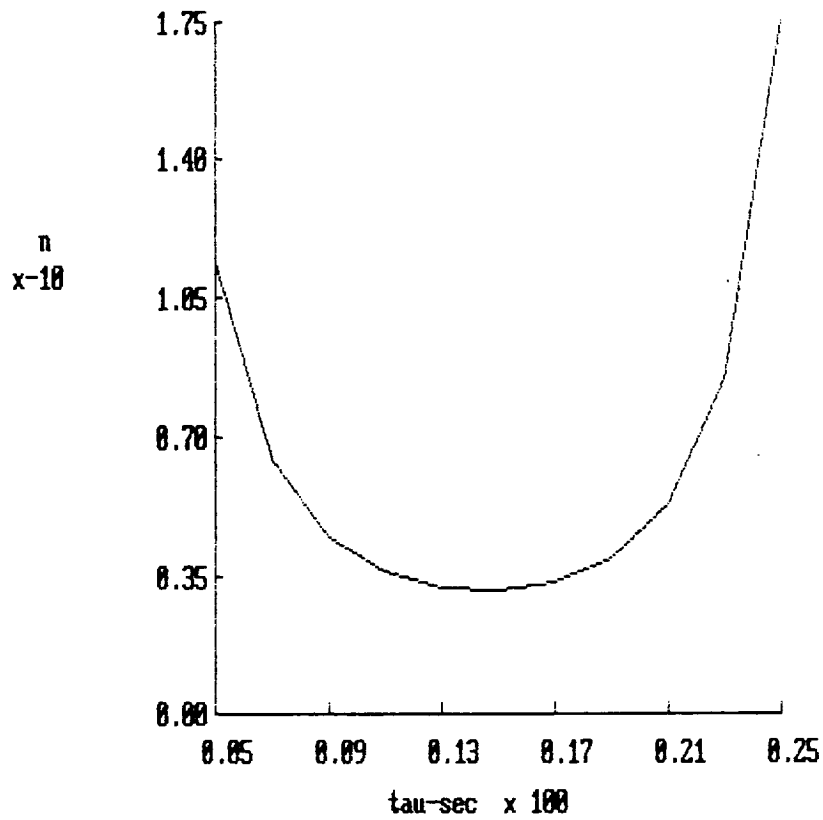
Specify which engine you wish to view
Enter 1 - 3 or 0 to continue 2

Sample Run

12:35PM 12-12-91

Engine No. 2

frequency = 110.000 Hertz



Specify which engine you wish to view
Enter 1 - 3 or 0 to continue 3

Sample Run

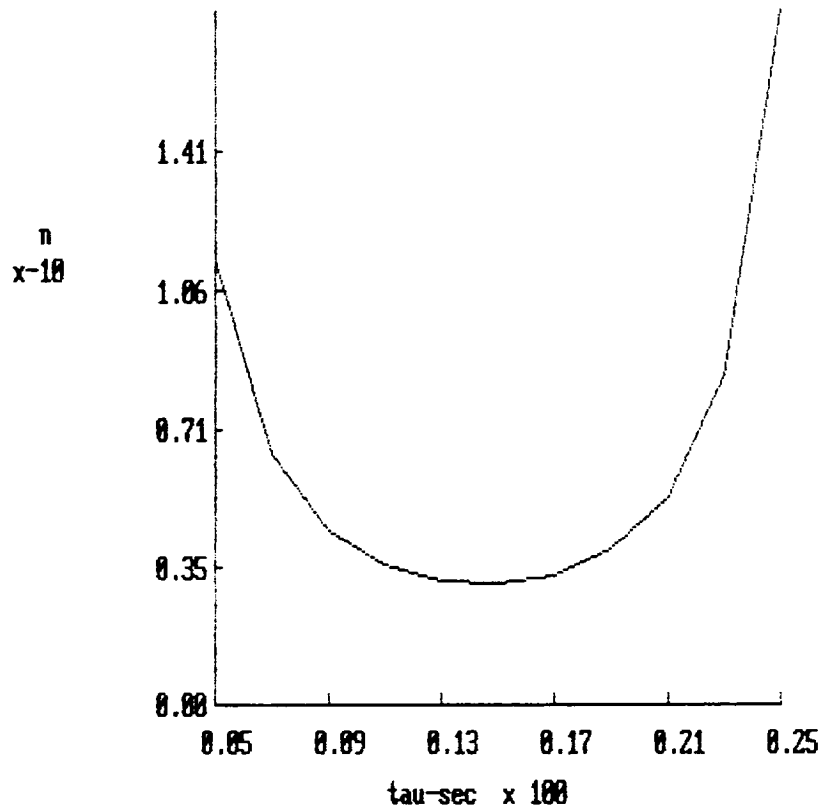
12:35PM

12-12-91

Engine No. 3

frequency =

110.000 Hertz



Specify which engine you wish to view
Enter 1 - 3 or 0 to continue 4
Invalid engine number, try again!
Specify which engine you wish to view
Enter 1 - 3 or 0 to continue 0

FREQUENCY = 1.13000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.08414E+01	4.69661E-07	-4.35565E-01
2	5.00000E-04	1.08143E+01	5.71385E-09	-4.32209E-01
3	5.00000E-04	1.08053E+01	-1.54009E-06	-4.31091E-01
1	7.00000E-04	6.16340E+00	-1.62474E-06	-2.79945E-01
2	7.00000E-04	6.12335E+00	-7.68963E-07	-2.76068E-01
3	7.00000E-04	6.11010E+00	1.38288E-06	-2.74779E-01
1	9.00000E-04	4.31322E+00	6.75733E-07	-1.80939E-01
2	9.00000E-04	4.26871E+00	1.11379E-06	-1.77752E-01
3	9.00000E-04	4.25390E+00	5.02842E-07	-1.76687E-01
1	1.10000E-03	3.47502E+00	-2.45920E-07	-1.05288E-01
2	1.10000E-03	3.43012E+00	8.45579E-08	-1.03835E-01
3	1.10000E-03	3.41516E+00	4.99280E-07	-1.03347E-01
1	1.30000E-03	3.12737E+00	-1.32093E-07	-3.97969E-02
2	1.30000E-03	3.08554E+00	1.74440E-07	-4.07200E-02
3	1.30000E-03	3.07158E+00	3.38107E-09	-4.10231E-02
1	1.50000E-03	3.10466E+00	3.06501E-08	2.23420E-02
2	1.50000E-03	3.07029E+00	1.45587E-08	1.90551E-02
3	1.50000E-03	3.05879E+00	-2.11063E-08	1.79628E-02
1	1.70000E-03	3.39810E+00	3.86204E-08	8.61902E-02
2	1.70000E-03	3.37799E+00	5.28391E-10	8.13690E-02
3	1.70000E-03	3.37122E+00	2.30528E-08	7.97620E-02
1	1.90000E-03	4.14782E+00	-7.43721E-09	1.58042E-01
2	1.90000E-03	4.15420E+00	2.49500E-08	1.53366E-01
3	1.90000E-03	4.15623E+00	4.26392E-09	1.51804E-01
1	2.10000E-03	5.81071E+00	6.40742E-09	2.49788E-01
2	2.10000E-03	5.86935E+00	5.01835E-09	2.47722E-01
3	2.10000E-03	5.88877E+00	-1.84221E-09	2.47025E-01
1	2.30000E-03	9.95287E+00	-5.58452E-09	3.91053E-01
2	2.30000E-03	1.01320E+01	-3.96782E-08	3.94885E-01
3	2.30000E-03	1.01916E+01	1.64012E-08	3.96150E-01
1	2.50000E-03	2.45212E+01	5.32486E-08	6.85143E-01
2	2.50000E-03	2.50927E+01	-2.75042E-08	7.00201E-01
3	2.50000E-03	2.52830E+01	-1.21638E-08	7.05209E-01

Do you wish to see n vs tau for this frequency? n

FREQUENCY = 1.16000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.02991E+01	-5.12383E-06	-4.01431E-01
2	5.00000E-04	1.02566E+01	-7.26954E-07	-4.02968E-01
3	5.00000E-04	1.02423E+01	2.17450E-06	-4.03475E-01
1	7.00000E-04	5.81341E+00	5.80983E-08	-2.41305E-01

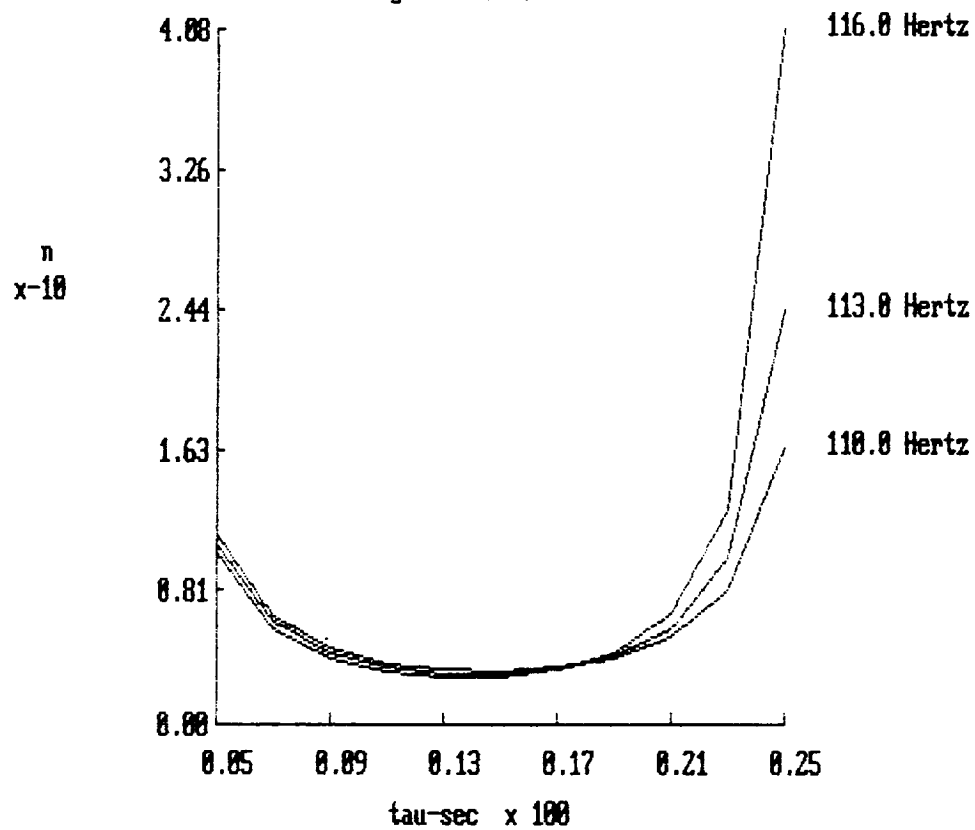
2	7.00000E-04	5.82060E+00	5.68050E-07	-2.45578E-01
3	7.00000E-04	5.82285E+00	-8.08546E-07	-2.46997E-01
1	9.00000E-04	4.04158E+00	1.28387E-06	-1.41208E-01
2	9.00000E-04	4.07478E+00	-5.29651E-07	-1.46441E-01
3	9.00000E-04	4.08578E+00	-4.13673E-07	-1.48185E-01
1	1.10000E-03	3.24865E+00	-5.17933E-08	-6.79342E-02
2	1.10000E-03	3.29868E+00	3.70957E-07	-7.22060E-02
3	1.10000E-03	3.31530E+00	2.70439E-07	-7.36336E-02
1	1.30000E-03	2.94248E+00	6.76098E-07	-7.44052E-03
2	1.30000E-03	3.00384E+00	-3.21188E-08	-8.97935E-03
3	1.30000E-03	3.02428E+00	1.41335E-07	-9.49907E-03
1	1.50000E-03	2.97883E+00	-4.58275E-08	4.91058E-02
2	1.50000E-03	3.04639E+00	-2.01901E-09	5.14330E-02
3	1.50000E-03	3.06893E+00	3.29763E-08	5.22016E-02
1	1.70000E-03	3.38149E+00	1.35664E-08	1.10026E-01
2	1.70000E-03	3.44778E+00	2.08757E-09	1.16225E-01
3	1.70000E-03	3.46994E+00	-9.54488E-09	1.18287E-01
1	1.90000E-03	4.36821E+00	-3.11257E-09	1.86303E-01
2	1.90000E-03	4.41867E+00	-1.24407E-08	1.94871E-01
3	1.90000E-03	4.43563E+00	1.78421E-09	1.97729E-01
1	2.10000E-03	6.64360E+00	-5.92199E-09	2.97441E-01
2	2.10000E-03	6.64246E+00	1.34157E-08	3.05073E-01
3	2.10000E-03	6.64233E+00	2.06930E-08	3.07628E-01
1	2.30000E-03	1.28961E+01	2.52306E-08	4.92079E-01
2	2.30000E-03	1.27256E+01	1.19181E-08	4.93000E-01
3	2.30000E-03	1.26692E+01	-3.55738E-08	4.93328E-01
1	2.50000E-03	4.07941E+01	-6.08384E-08	9.70344E-01
2	2.50000E-03	3.97816E+01	-7.47497E-08	9.51817E-01
3	2.50000E-03	3.94452E+01	-1.56099E-08	9.45674E-01

Do you wish to see n vs tau for this frequency? n

Sample Run

12:35PM 12-12-91

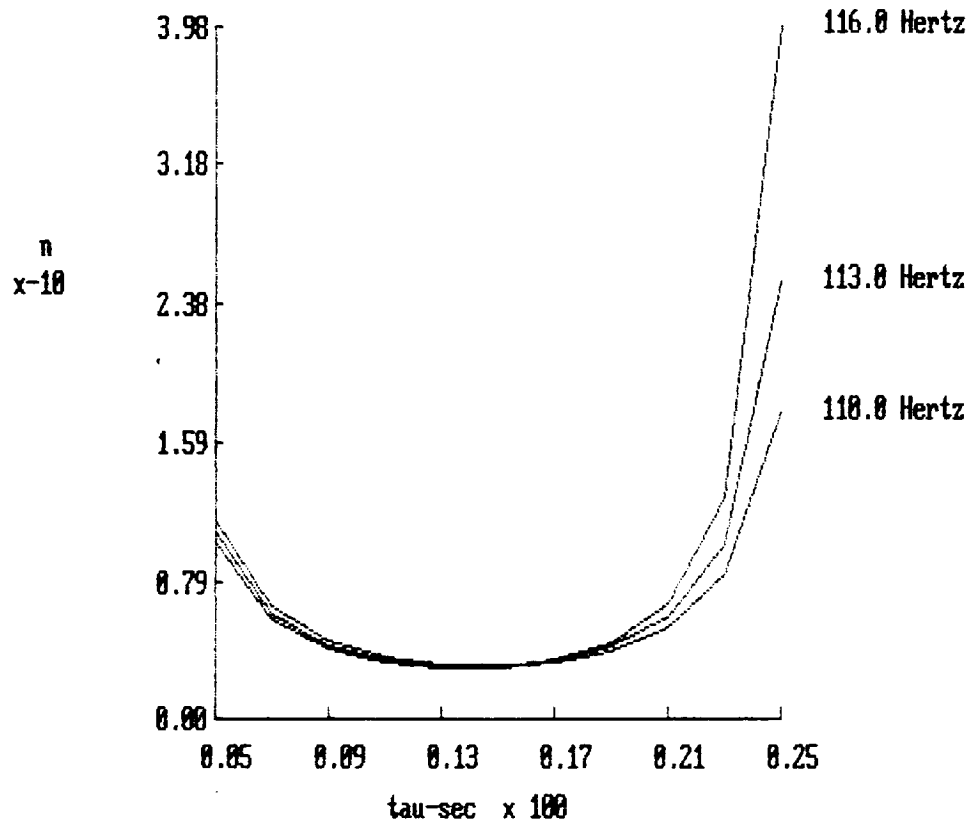
Engine No. 1



Sample Run

12:35PM 12-12-91

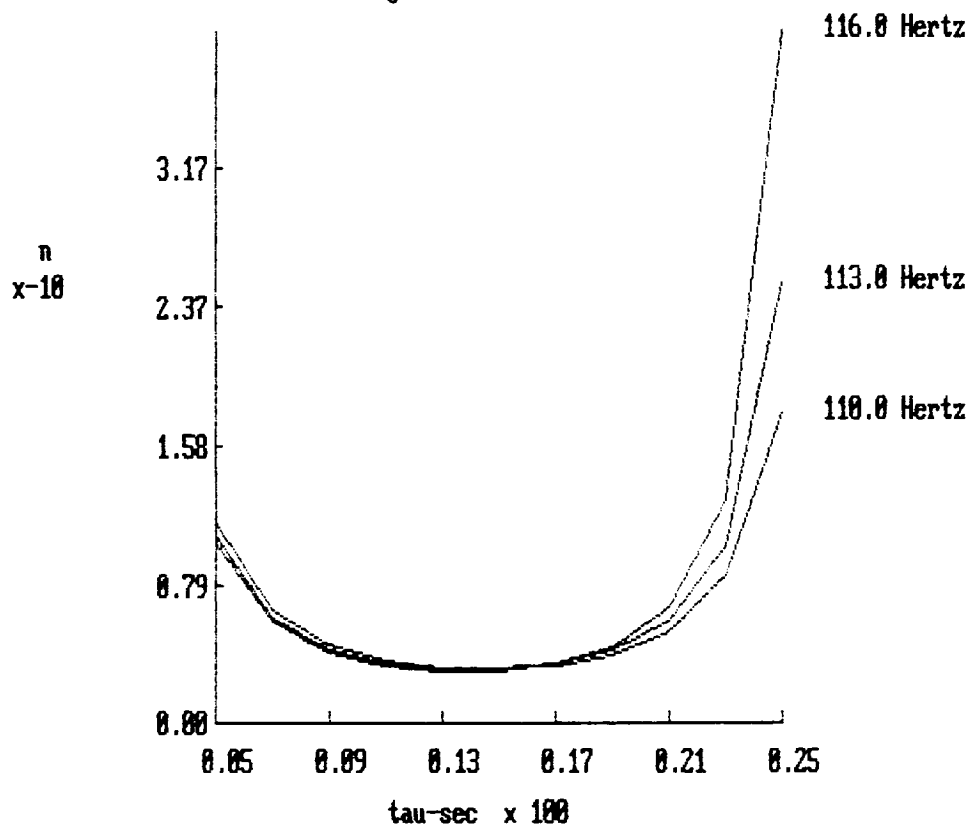
Engine No. 2



Sample Run

12:35PM 12-12-91

Engine No. 3



Do you want to run another case? Enter Y or N n

4.3 Output for Sample Run

IMODE.OUT File

CAREA= 8.113004
TAREA= 3.912716
ASTAR= 2869.983000
CSTARD= 66403.230000
RHOBAR= 2.110266
UBARD= 49.852240
RHOBAR= 2.110266
UBARD= 49.852240

Sample Run

12:35PM 12-12-91

Engine No. 1

DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 4.00000E+00
UBAR= 4.98522E+01 4.98522E+01
DTAU = 6.97000E-04 NR = 1.00000E-02 CDIAM = 3.21400E+00
TDIAM = 2.23200E+00 XLC = 4.00000E+00 GAMMA = 1.20000E+00
RGAS = 1.71600E+03 P00 = 1.42500E+05 MBAR = 8.53500E+02
RBAR = 2.67000E+00 DCSDR = -3.15000E+02 DHLDR = 1.00000E-02
RHOLO = 4.40000E-01 ULO = 1.96500E+03 PCHMB = 4.50204E+05
TCHMB = 4.00000E+03

NON-DIMENSIONAL VARIABLES

NVAL= 2
XBAR= 0.00000E+00 1.00000E+00
UBAR= 1.73702E-02 1.73702E-02
DTAU= 5.00094E-01 NR= 1.00000E-02 RBAR= 2.67000E+00
MBAR= 6.94809E-02 GAMMA= 1.20000E+00 P00= 3.16523E-01
DHLDR= 1.00000E-02 CSTAR= 2.31372E+01 DCSDR= -1.09757E-01
RHOLO= 2.08505E-01 ULO= 6.84673E-01
RFA= 1.44752E-03 0.00000E+00 RFC= 0.00000E+00 0.00000E+00

CAREA= 8.113004
 TAREA= 3.912716
 ASTAR= 2869.983000
 CSTARD= 33201.620000
 RHOBAR= 2.110266
 UBARD= 99.704480
 RHOBAR= 2.110266
 UBARD= 99.704480

Sample Run

12:35PM 12-12-91

Engine No. 2

DIMENSIONAL VARIABLES

NVAL= 2
 XBAR= 0.00000E+00 4.00000E+00
 UBAR= 9.97045E+01 9.97045E+01
 DTAU = 6.97000E-04 NR = 1.00000E-02 CDIAM = 3.21400E+00
 TDIAM = 2.23200E+00 XLC = 4.00000E+00 GAMMA = 1.20000E+00
 RGAS = 1.71600E+03 P00 = 1.42500E+05 MBAR = 1.70700E+03
 RBAR = 2.67000E+00 DCSDR = -3.15000E+02 DHLDLDR = 1.00000E-02
 RHOLO = 4.40000E-01 ULO = 1.96500E+03 PCHMB = 4.50204E+05
 TCHMB = 4.00000E+03

NON-DIMENSIONAL VARIABLES

NVAL= 2
 XBAR= 0.00000E+00 1.00000E+00
 UBAR= 3.47404E-02 3.47404E-02
 DTAU= 5.00094E-01 NR= 1.00000E-02 RBAR= 2.67000E+00
 MBAR= 1.38962E-01 GAMMA= 1.20000E+00 P00= 3.16523E-01
 DHLDLDR= 1.00000E-02 CSTAR= 1.15686E+01 DCSDR= -1.09757E-01
 RHOLO= 2.08505E-01 ULO= 6.84673E-01
 RFA= 2.89504E-03 0.00000E+00 RFC= 0.00000E+00 0.00000E+00

CAREA= 8.113004
 TAREA= 3.912716
 ASTAR= 2869.983000
 CSTARD= 16600.810000
 RHOBAR= 2.110266
 UBARD= 199.409000
 RHOBAR= 2.110266
 UBARD= 199.409000

Sample Run

12:35PM 12-12-91

Engine No. 3

DIMENSIONAL VARIABLES

NVAL= 2
 XBAR= 0.00000E+00 4.00000E+00
 UBAR= 1.99409E+02 1.99409E+02
 DTAU = 6.97000E-04 NR = 1.00000E-02 CDIAM = 3.21400E+00
 TDIAM = 2.23200E+00 XLC = 4.00000E+00 GAMMA = 1.20000E+00
 RGAS = 1.71600E+03 P00 = 1.42500E+05 MBAR = 3.41400E+03
 RBAR = 2.67000E+00 DCSDR = -3.15000E+02 DHLDLDR = 1.00000E-02
 RHOLO = 4.40000E-01 ULO = 1.96500E+03 PCHMB = 4.50204E+05
 TCHMB = 4.00000E+03

NON-DIMENSIONAL VARIABLES

NVAL= 2
 XBAR= 0.00000E+00 1.00000E+00
 UBAR= 6.94809E-02 6.94809E-02
 DTAU= 5.00094E-01 NR= 1.00000E-02 RBAR= 2.67000E+00
 MBAR= 2.77924E-01 GAMMA= 1.20000E+00 P00= 3.16523E-01
 DHLDLDR= 1.00000E-02 CSTAR= 5.78429E+00 DCSDR= -1.09757E-01
 RHOLO= 2.08505E-01 ULO= 6.84673E-01
 RFA= 5.79008E-03 0.00000E+00 RFC= 0.00000E+00 0.00000E+00

Sample Run

12:35PM 12-12-91

FREQUENCY = 1.10000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.14621E+01	-3.11896E-07	-4.67076E-01
2	5.00000E-04	1.14841E+01	-3.03067E-07	-4.60113E-01
3	5.00000E-04	1.14920E+01	3.88765E-07	-4.57809E-01
1	7.00000E-04	6.55967E+00	8.66792E-07	-3.17231E-01
2	7.00000E-04	6.48823E+00	-2.23960E-06	-3.05747E-01
3	7.00000E-04	6.46472E+00	-1.18901E-06	-3.01928E-01
1	9.00000E-04	4.62124E+00	2.49193E-07	-2.20846E-01
2	9.00000E-04	4.50544E+00	6.42385E-07	-2.08742E-01
3	9.00000E-04	4.46697E+00	-1.25355E-06	-2.04704E-01
1	1.10000E-03	3.73689E+00	-2.28634E-07	-1.44484E-01
2	1.10000E-03	3.59644E+00	-2.45890E-07	-1.35610E-01
3	1.10000E-03	3.54970E+00	-7.22797E-07	-1.32642E-01
1	1.30000E-03	3.35282E+00	1.55936E-07	-7.53268E-02
2	1.30000E-03	3.20046E+00	1.72578E-07	-7.29305E-02
3	1.30000E-03	3.14967E+00	-2.02976E-07	-7.21169E-02
1	1.50000E-03	3.28318E+00	-4.40167E-08	-7.92099E-03
2	1.50000E-03	3.13166E+00	5.27691E-08	-1.37797E-02
3	1.50000E-03	3.08109E+00	5.24031E-08	-1.57178E-02
1	1.70000E-03	3.49309E+00	-2.72676E-09	6.03049E-02
2	1.70000E-03	3.36009E+00	-3.73278E-09	4.66884E-02
3	1.70000E-03	3.31559E+00	-2.31758E-08	4.21583E-02
1	1.90000E-03	4.06690E+00	2.46680E-08	1.31826E-01
2	1.90000E-03	3.98268E+00	1.20805E-08	1.13762E-01
3	1.90000E-03	3.95433E+00	-1.35946E-08	1.07738E-01
1	2.10000E-03	5.30008E+00	-2.17681E-08	2.12487E-01
2	2.10000E-03	5.32637E+00	-6.04931E-09	1.96344E-01
3	2.10000E-03	5.33471E+00	-1.82410E-08	1.90944E-01
1	2.30000E-03	8.12702E+00	-2.26766E-08	3.19093E-01
2	2.30000E-03	8.41822E+00	-8.19192E-09	3.14610E-01
3	2.30000E-03	8.51465E+00	-2.97111E-08	3.13079E-01
1	2.50000E-03	1.64494E+01	-1.18208E-10	5.06763E-01
2	2.50000E-03	1.75388E+01	3.19730E-08	5.29430E-01
3	2.50000E-03	1.79009E+01	3.66275E-10	5.36935E-01

Sample Run

12:35PM 12-12-91

FREQUENCY = 1.13000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.08414E+01	4.69661E-07	-4.35565E-01
2	5.00000E-04	1.08143E+01	5.71385E-09	-4.32209E-01
3	5.00000E-04	1.08053E+01	-1.54009E-06	-4.31091E-01
1	7.00000E-04	6.16340E+00	-1.62474E-06	-2.79945E-01
2	7.00000E-04	6.12335E+00	-7.68963E-07	-2.76068E-01
3	7.00000E-04	6.11010E+00	1.38288E-06	-2.74779E-01
1	9.00000E-04	4.31322E+00	6.75733E-07	-1.80939E-01
2	9.00000E-04	4.26871E+00	1.11379E-06	-1.77752E-01
3	9.00000E-04	4.25390E+00	5.02842E-07	-1.76687E-01
1	1.10000E-03	3.47502E+00	-2.45920E-07	-1.05288E-01
2	1.10000E-03	3.43012E+00	8.45579E-08	-1.03835E-01
3	1.10000E-03	3.41516E+00	4.99280E-07	-1.03347E-01
1	1.30000E-03	3.12737E+00	-1.32093E-07	-3.97969E-02
2	1.30000E-03	3.08554E+00	1.74440E-07	-4.07200E-02
3	1.30000E-03	3.07158E+00	3.38107E-09	-4.10231E-02
1	1.50000E-03	3.10466E+00	3.06501E-08	2.23420E-02
2	1.50000E-03	3.07029E+00	1.45587E-08	1.90551E-02
3	1.50000E-03	3.05879E+00	-2.11063E-08	1.79628E-02
1	1.70000E-03	3.39810E+00	3.86204E-08	8.61902E-02
2	1.70000E-03	3.37799E+00	5.28391E-10	8.13690E-02
3	1.70000E-03	3.37122E+00	2.30528E-08	7.97620E-02
1	1.90000E-03	4.14782E+00	-7.43721E-09	1.58042E-01
2	1.90000E-03	4.15420E+00	2.49500E-08	1.53366E-01
3	1.90000E-03	4.15623E+00	4.26392E-09	1.51804E-01
1	2.10000E-03	5.81071E+00	6.40742E-09	2.49788E-01
2	2.10000E-03	5.86935E+00	5.01835E-09	2.47722E-01
3	2.10000E-03	5.88877E+00	-1.84221E-09	2.47025E-01
1	2.30000E-03	9.95287E+00	-5.58452E-09	3.91053E-01
2	2.30000E-03	1.01320E+01	-3.96782E-08	3.94885E-01
3	2.30000E-03	1.01916E+01	1.64012E-08	3.96150E-01
1	2.50000E-03	2.45212E+01	5.32486E-08	6.85143E-01
2	2.50000E-03	2.50927E+01	-2.75042E-08	7.00201E-01
3	2.50000E-03	2.52830E+01	-1.21638E-08	7.05209E-01

Sample Run

12:35PM 12-12-91

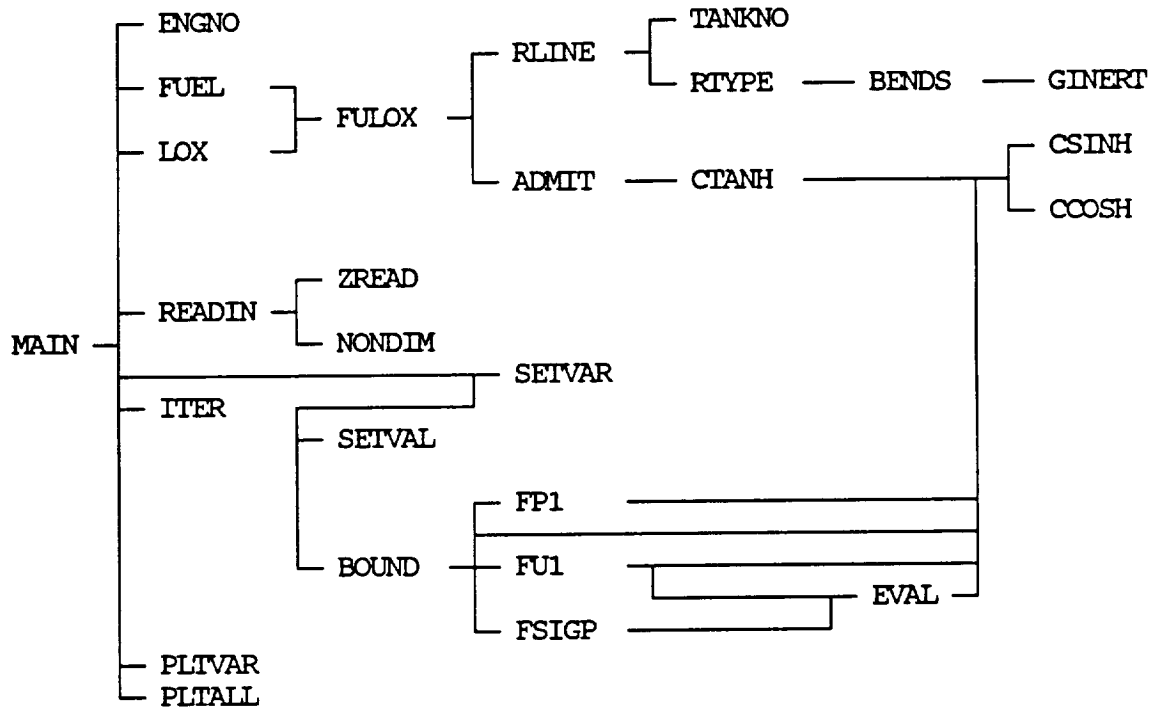
FREQUENCY = 1.16000E+02 Hertz

Eng. No.	tau-sec	n	FUNB(R)	FUNB(I)
1	5.00000E-04	1.02991E+01	-5.12383E-06	-4.01431E-01
2	5.00000E-04	1.02566E+01	-7.26954E-07	-4.02968E-01
3	5.00000E-04	1.02423E+01	2.17450E-06	-4.03475E-01
1	7.00000E-04	5.81341E+00	5.80983E-08	-2.41305E-01
2	7.00000E-04	5.82060E+00	5.68050E-07	-2.45578E-01
3	7.00000E-04	5.82285E+00	-8.08546E-07	-2.46997E-01
1	9.00000E-04	4.04158E+00	1.28387E-06	-1.41208E-01
2	9.00000E-04	4.07478E+00	-5.29651E-07	-1.46441E-01
3	9.00000E-04	4.08578E+00	-4.13673E-07	-1.48185E-01
1	1.10000E-03	3.24865E+00	-5.17933E-08	-6.79342E-02
2	1.10000E-03	3.29868E+00	3.70957E-07	-7.22060E-02
3	1.10000E-03	3.31530E+00	2.70439E-07	-7.36336E-02
1	1.30000E-03	2.94248E+00	6.76098E-07	-7.44052E-03
2	1.30000E-03	3.00384E+00	-3.21188E-08	-8.97935E-03
3	1.30000E-03	3.02428E+00	1.41335E-07	-9.49907E-03
1	1.50000E-03	2.97883E+00	-4.58275E-08	4.91058E-02
2	1.50000E-03	3.04639E+00	-2.01901E-09	5.14330E-02
3	1.50000E-03	3.06893E+00	3.29763E-08	5.22016E-02
1	1.70000E-03	3.38149E+00	1.35664E-08	1.10026E-01
2	1.70000E-03	3.44778E+00	2.08757E-09	1.16225E-01
3	1.70000E-03	3.46994E+00	-9.54488E-09	1.18287E-01
1	1.90000E-03	4.36821E+00	-3.11257E-09	1.86303E-01
2	1.90000E-03	4.41867E+00	-1.24407E-08	1.94871E-01
3	1.90000E-03	4.43563E+00	1.78421E-09	1.97729E-01
1	2.10000E-03	6.64360E+00	-5.92199E-09	2.97441E-01
2	2.10000E-03	6.64246E+00	1.34157E-08	3.05073E-01
3	2.10000E-03	6.64233E+00	2.06930E-08	3.07628E-01
1	2.30000E-03	1.28961E+01	2.52306E-08	4.92079E-01
2	2.30000E-03	1.27256E+01	1.19181E-08	4.93000E-01
3	2.30000E-03	1.26692E+01	-3.55738E-08	4.93328E-01
1	2.50000E-03	4.07941E+01	-6.08384E-08	9.70344E-01
2	2.50000E-03	3.97816E+01	-7.47497E-08	9.51817E-01
3	2.50000E-03	3.94452E+01	-1.56099E-08	9.45674E-01

SURF.ERR File

None created for this frequency range.

5.0 Flow Diagram



6.0 Variable Description

Variables in Commons

/CMPVAL/		
CVAR(17)	COMPLEX*8	equivalence(CVAR(1),X1)
X1	COMPLEX*8	first order term of x
Y1	COMPLEX*8	first order term of y
Z1	COMPLEX*8	first order term of z
W1	COMPLEX*8	first order term of w
M1	COMPLEX*8	first order term of m
P0	COMPLEX*8	zeroth order term of pressure
P1	COMPLEX*8	first order term of pressure
U0	COMPLEX*8	zeroth order term of velocity
U1	COMPLEX*8	first order term of velocity
RFH	COMPLEX*8	comb. response function for mixture ratio
RFK	COMPLEX*8	comb. response function for mass flow
RFP	COMPLEX*8	comb. response function for pressure
S	COMPLEX*8	lamda + mu I - perturbation oscillation
GF	COMPLEX*8	admittance of fuel line looking toward tank
GOX	COMPLEX*8	admittance of lox line looking toward tank
RFA	COMPLEX*8	nozzle pressure admittance coefficient
RFC	COMPLEX*8	nozzle entropy admittance coefficient
/DIMVAL/		
HOLDD(20)	REAL*4	equivalence(HOLDD(1),ND)
ND	REAL*4	pressure interaction index
TAUD	REAL*4	sensitive time lag (sec)
DTAUD	REAL*4	delta time lag (sec)
NRD	REAL*4	mixture ratio interaction index
LAMDAD	REAL*4	damping of perturbation
MUD	REAL*4	frequency of perturbation (rad/sec)
CDIAM	REAL*4	chamber diameter (ft)
TDIAM	REAL*4	throat diameter (ft)
XLCD	REAL*4	x location of chamber-nozzle interface (ft)
GAMMAD	REAL*4	ratio of specific heats
RGAS	REAL*4	gas constant (ft ² /sec ² /°R)
POOD	REAL*4	maximum pressure at injection face (lbf/ft ²)
MBARD	REAL*4	mean comb. response function (lbm/sec)
RBARD	REAL*4	mean mixture ratio
DCSDRD	REAL*4	d(cstar)/d(mixture ratio) (ft/sec)
DHLDRD	REAL*4	d(enthalpy)/d(mixture ratio) (ft ² /sec ²)
RHOLOD	REAL*4	mass of liquid per unit chamber vol (lbm/ft ³)
ULOD	REAL*4	axial component of liquid velocity (ft/sec)
PCHMB	REAL*4	chamber pressure (lbf/ft ²)
TCHMB	REAL*4	chamber temperature (°R)
XBARD(50)	REAL*4	x locations along axis (ft)
PBAR(50)	REAL*4	pressure along axis (lbf/ft ²)
TBAR(50)	REAL*4	temperature along axis (°R)


```

                                /EPARAM/
MENG                INTEGER*2  number of engines
TFLOW(25)           REAL*4     total flow rate of engine (lbm/sec)
PCHMB(25)           REAL*4     chamber pressure (lbf/ft^2)
DPROR(25)           REAL*4     pressure drop across orifices (lbf/ft^2)
PMRAT(25)           REAL*4     chamber pressure/total mass flow

                                /FOPIPE/
PIPE1(75,25)        REAL*4     first parameter of pipe description
PIPE2(75,25)        REAL*4     second parameter of pipe description
PIPE3(75,25)        REAL*4     third parameter of pipe description
PIPE4(75,25)        REAL*4     fourth parameter of pipe description
PIPE5(75,25)        REAL*4     fifth parameter of pipe description

                                /FPARAM/
MLINE               INTEGER*2  number of lines from tank
SPLIT(25)           REAL*4     number of unique lines from pipe split
A(25)               REAL*4     speed of sound in the fluid (ft/sec)
CMAN(25)            REAL*4     manifold capacitance
CTANK(25)           REAL*4     tank capacitance
DENS(25)            REAL*4     density of fluid (lbm/ft^3)
KMAN(25)            REAL*4     bulk modulus of manifold (lbf/ft^2)
KTANK(25)           REAL*4     bulk modulus of tank (lbf/ft^2)
LFLOW(25)           REAL*4     flow rate through pipe (lbm/sec)
VOL(25)            REAL*4     volume of tank (ft^3)
VOLMF(25)           REAL*4     volume of manifold (ft^3)
AREA(75,25)         REAL*4     area of pipe section (ft^2)
DIA(75,25)          REAL*4     diameter of pipe section (ft)
L(75,25)            REAL*4     length of pipe section (ft)
PIND(75,25)         REAL*4     inductance of pipe section
PCAP(75,25)         REAL*4     capacitance of pipe section
AVGK(25)            REAL*4     average bulk modulus
SEGMNF(25)          INTEGER*2  number of pipe sections
SECTNF(75,25)       INTEGER*2  pipe section type
NOLINE(25)          INTEGER*2  number of identical lines
IENG(25)            INTEGER*2  engine number
ITANK(25)           INTEGER*2  tank number
LOPOLD(25)          INTEGER*2  previous maximum number of iterations
LOPEND(25)          INTEGER*2  maximum number of iterations for split pipe

                                /INTVAL/
NVAL                INTEGER*2  number of points along chamber

                                /OPARAM/
MLINE               INTEGER*2  number of lines from tank
SPLIT(25)           REAL*4     number of unique lines from pipe split
A(25)               REAL*4     speed of sound in the fluid (ft/sec)
CMAN(25)            REAL*4     manifold capacitance
CTANK(25)           REAL*4     tank capacitance
DENS(25)            REAL*4     density of fluid (lbm/ft^3)
KMAN(25)            REAL*4     bulk modulus of manifold (lbf/ft^2)
KTANK(25)           REAL*4     bulk modulus of tank (lbf/ft^2)
LFLOW(25)           REAL*4     flow rate through pipe (lbm/sec)

```

VOL(25)	REAL*4	volume of tank (ft ³)
VOLMF(25)	REAL*4	volume of manifold (ft ³)
AREA(75,25)	REAL*4	area of pipe section (ft ²)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
L(75,25)	REAL*4	length of pipe section (ft)
PIND(75,25)	REAL*4	inductance of pipe section
PCAP(75,25)	REAL*4	capacitance of pipe section
AVGK(25)	REAL*4	average bulk modulus
SEGMNF(25)	INTEGER*2	number of pipe sections
SECTNF(75,25)	INTEGER*2	pipe section type
NOLINE(25)	INTEGER*2	number of identical lines
IENG(25)	INTEGER*2	engine number
ITANK(25)	INTEGER*2	tank number
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe

/PIPES/

PFACE	REAL*4	pressure at injector face (lbf/ft ²)
TFACE	REAL*4	mean comb. response function (lkm/sec)
ASTAR	REAL*4	speed of sound at injector face (ft/sec)

/RELVAL/

RVAR(13)	REAL*4	equivalence(RVAR(1),N)
N	REAL*4	pressure interaction index
TAU	REAL*4	sensitive time lag
DTAU	REAL*4	delta time lag
NR	REAL*4	mixture ratio interaction index
RBAR	REAL*4	mean mixture ratio
MBAR	REAL*4	mean comb. response function
GAMMA	REAL*4	ratio of specific heats
P00	REAL*4	maximum pressure at injection face
DHLDR	REAL*4	d(enthalpy)/d(mixture ratio)
CSTAR	REAL*4	characteristic velocity at combustor exit
DCSDR	REAL*4	d(cstar)/d(mixture ratio)
RHOLO	REAL*4	mass of liquid per unit chamber volume
ULO	REAL*4	axial component of liquid velocity
LAMDA	REAL*4	damping of perturbation
MU	REAL*4	frequency of perturbation
TAUT	REAL*4	total time lag
UBAR(50)	REAL*4	velocity along axis
XBAR(50)	REAL*4	x locations along axis
XLC	REAL*4	x location of chamber-nozzle interface

/RESULT/

PP	COMPLEX*8	$P' = P_0 + P_1$
UP	COMPLEX*8	$U' = U_0 + U_1$
SIGP	COMPLEX*8	$SIG' = SIG_0 + SIG_1$
FUNB	COMPLEX*8	boundary function $U' + RFA * P' + RFC * SIG'$

/SFAC/

SFAC	REAL*4	factor for frequency
------	--------	----------------------

		/TTTL/
TITLE	CHAR*60	title for plots including date and time
TTTLF	CHAR*40	input title
IHR	INTEGER*2	hour code run
IMIN	INTEGER*2	minute code run
AP	CHAR*2	AM or PM
IYR	INTEGER*2	yesr code run
IMON	INTEGER*2	month code run
IDAY	INTEGER*2	day code run

		/WCAOUT/
NAMLIN(2)	CHAR*24	name of files containing pipe description
IUNITH	INTEGER*2	unit number of current file

		/WORK/
YP(50,50)	REAL*4	n array
YP1(20,25)	REAL*4	work array
YP2(50,25)	REAL*4	work array
YP3(50,25)	REAL*4	work array
YPA(50,50)	REAL*4	n work array
YP4(50,25)	REAL*4	work array
YP5(30,25)	REAL*4	work array

		/WORK1/
G(0:75,25)	COMPLEX*8	admittance looking toward tank
ZT(0:75,25)	COMPLEX*8	impedance looking toward tank
ZG(0:75,25)	COMPLEX*8	impedance looking toward engine

		/WORK2/
ZO(75,25)	REAL*4	characteristic impedance

PROGRAM SSFREQ
 Logic portion of code

Commons	CMPVAL	DIMVAL	EPARAM	INTVAL	RELVAL	RESULT	SFACT
	TTTL	WORK					
		Local Variables					
AM	CHAR*2	'AM'					
ANS	CHAR*1	response to question					
DELF	REAL*4	intermediate variable					
DELVAL	REAL*4	intermediate variable					
FREQ(50)	REAL*4	array of frequencies					
GFA(25)	COMPLEX*8	fuel line admittance					
GOXA(25)	COMPLEX*8	lox line admittance					
I	INTEGER*2	do loop index					
IFREQ	INTEGER*2	flag denoting presence of frequency file					
IGONE	INTEGER*2	flag for subroutine fuel or lox					
ISEC	INTEGER*2	seconds at start					
ITAU	INTEGER*2	flag denoting presence of tau file					
I100	INTEGER*2	hundreds of seconds at start					
J	INTEGER*2	do loop index					
JUNIT	INTEGER*2	unit number of engine data file					

K	INTEGER*2	do loop index
NA(25)	REAL*4	n for each engine
NAMENG	CHAR*24	name of engine file
NOF	INTEGER*2	maximum number of frequencies
NOT	INTEGER*2	maximum number of tau's
NPTF	INTEGER*2	number of frequencies
NPTS	INTEGER*2	number of tau's
PM	CHAR*2	'PM'
RADHER(2)	CHAR*8	labels
ROCIN	CHAR*24	input file name
ROCVAR	CHAR*24	file name for frequencies or tau's
STARTF	REAL*4	starting frequency
STARTV	REAL*4	starting tau
STOPF	REAL*4	ending frequency
STOPV	REAL*4	ending tau
TAULST(200)	REAL*4	array of tau's
TOL	REAL*4	convergence criteria
VARF	REAL*4	current frequency
VARP(3)	CHAR*8	labels
VART	REAL*4	current tau
VAR1	REAL*4	intermediate variable

SUBROUTINE ADMITT

Determines admittance looking toward tank

Commons	FACTOR	TITL	WORK1	WORK2
Variables in Argument List				
A	REAL*4	speed of sound in the fluid (ft/sec)		
AREA(75,25)	REAL*4	area of pipe section (ft ²)		
CMAN(25)	REAL*4	manifold capacitance		
CTANK	REAL*4	tank capacitance		
DPROR(25)	REAL*4	pressure drop across orifices (lbf/ft ²)		
GADM(25)	COMPLEX*8	admittance looking toward tank		
IENG(25)	INTEGER*2	engine number		
IP	INTEGER*2	current pipe section		
ITLIN	INTEGER*2	flag for fuel or lox		
L(75,25)	REAL*4	length of pipe section (ft)		
LFLOW	REAL*4	flow rate through pipe (lbm/sec)		
LOPEND	INTEGER*2	maximum number of iterations for split pipe		
NOLINE(25)	INTEGER*2	number of identical lines		
PCAP(75,25)	REAL*4	capacitance of pipe section		
PIND(75,25)	REAL*4	inductance of pipe section		
PMRAT(25)	REAL*4	chamber pressure/total mass flow		
S	COMPLEX*8	current frequency		
SECTIN(75,25)	INTEGER*2	pipe section type		
SEGMN(25)	INTEGER*2	number of pipe sections		
SPLIT	REAL*4	number of unique lines from pipe split		
TFLOW(25)	REAL*4	total flow rate of engine (lbm/sec)		
Local Variables				
CAPM	COMPLEX*8	intermediate variable		
CAPN	COMPLEX*8	intermediate variable		
CFAC	COMPLEX*8	intermediate variable		

ERRP	REAL*4	convergence error
GDIF	REAL*4	maximum difference in admittance
GOLD(0:75,25)	COMPLEX*8	previous admittance
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec ²)
I	INTEGER*2	do loop index
IE	INTEGER*2	current engine number
IEE	INTEGER*2	engine number
IOPEN	INTEGER*2	flag indicating if SURF.ERR is open
IWG	INTEGER*2	first index of maximum error
J	INTEGER*2	do loop index
JWG	INTEGER*2	second index of maximum error
K	INTEGER*2	do loop index
KLOOP	INTEGER*2	do loop index
LOPHI	INTEGER*2	intermediate variable
RATPM	REAL*4	intermediate variable
RHS	COMPLEX*8	intermediate variable
TCOUNT	REAL*4	intermediate variable
TL	REAL*4	length/speed of sound
TMASS	REAL*4	intermediate variable
TYPEL(2)	CHAR*13	intermediate array
WG	REAL*4	intermediate variable
WGOLD	REAL*4	intermediate variable
ZGEFF	COMPLEX*8	effective impedance for calculations
ZLP	REAL*4	intermediate variable
ZOEFF	REAL*4	effective Z0 for calculations
ZOR(25)	REAL*4	intermediate variable
ZTEFF	COMPLEX*8	effective Zt for calculations
ZTOP	REAL*4	intermediate variable

SUBROUTINE BENDS

Computes effective straight pipe for bend

Variables in Argument List		
DIME	REAL*4	effective diameter (ft)
PIPE1	REAL*4	radius of bend (ft)
PIPE2	REAL*4	angle of bend (degrees)
PIPE3	REAL*4	diameter of bend (ft)
PIPE4	REAL*4	length of end straight segments (ft)
VALUE	REAL*4	effective length (ft)
Local Variables		
GAMMA	REAL*4	intermediate variable
LBEND	REAL*4	intermediate variable
RATIO	REAL*4	intermediate variable
Y	REAL*4	intermediate variable

SUBROUTINE BOUND

Evaluates the boundary function

Commons CMPVAL INTVAL RELVAL

Variables in Argument List

FUNB	COMPLEX*8	boundary function $U' + RFA * P' + RFC * SIG'$
------	-----------	--

PP	COMPLEX*8	$P' = P_0 + P_1$
SIGP	COMPLEX*8	$SIG' = SIG_0 + SIG_1$
UP	COMPLEX*8	$U' = U_0 + U_1$

COMPLEX FUNCTION CCOSH
Evaluates the complex hyperbolic cosine

	Variables in Argument List	
S	COMPLEX*8	current frequency
	Local Variables	
COSHI	REAL*4	intermediate variable
COSHR	REAL*4	intermediate variable
LAMDA	REAL*4	real part of complex frequency
MU	REAL*4	complex part of complex frequency

COMPLEX FUNCTION CSINH
Evaluates the complex hyperbolic sine

	Variables in Argument List	
S	COMPLEX*8	current frequency
	Local Variables	
LAMDA	REAL*4	real part of complex frequency
MU	REAL*4	complex part of complex frequency
SINHI	REAL*4	intermediate variable
SINHR	REAL*4	intermediate variable

COMPLEX FUNCTION CTANH
Evaluates the complex hyperbolic tangent

	Variables in Argument List	
S	COMPLEX*8	current frequency
	Local Variables	
CTAND	COMPLEX*8	intermediate variable
CTANN	COMPLEX*8	intermediate variable

SUBROUTINE ENGNO
Reads engine parameters

Commons EPARAM		Variables in Argument List
IUNIT	INTEGER*2	unit number of engine file
	Local Variables	
I	INTEGER*2	do loop index

SUBROUTINE EVAL
Evaluates parameters at a given x location

Commons CMPVAL INTVAL RELVAL

	Variables in Argument List	
X	REAL*4	current x station
	Local Variables	
FAC	REAL*4	intermediate variable
I	INTEGER*2	do loop index
UB	REAL*4	intermediate variable

COMPLEX FUNCTION FP1
Evaluates P1

Commons	CMPVAL	INTVAL	RELVAL
	Variables in Argument List		
XL	REAL*4	length of chamber	
	Local Variables		
DX	REAL*4	integration increment	
I	INTEGER*2	do loop variable	
VINT	COMPLEX*8	intermediate variable	
X	REAL*4	current x location	

COMPLEX FUNCTION FSIGP
Evaluates SIG'

Commons	CMPVAL	INTVAL	RELVAL
	Variables in Argument List		
XL	REAL*4	length of chamber	
	Local Variables		
DX	REAL*4	integration increment	
FA	REAL*4	intermediate variable	
FOON	COMPLEX*8	intermediate variable	
FSIG2	COMPLEX*8	intermediate variable	
I	INTEGER*2	do loop index	
II	INTEGER*2	do loop index	
J	INTEGER*2	do loop index	
UB(51)	REAL*4	intermediate array	
VINT(51)	COMPLEX*8	intermediate array	
VVINT(51)	COMPLEX*8	intermediate array	
X	REAL*4	current x location	

SUBROUTINE FUEL
Handles fuel piping logic

Commons	EPARAM	FOPIPE	FPARAM	WCAOUT
	Variables in Argument List			
GF(25)	COMPLEX*8	admittance looking toward tank		
IGONE	INTEGER*2	flag for subroutine fuel or lox		
IUNIT	INTEGER*2	unit number of fuel data file		
IUNITP	INTEGER*2	unit number of fuel work file		
S	COMPLEX*8	current frequency		
	Local Variables			
ANS	CHAR*1	response to question		

FUELIN CHAR*24 name of fuel data file

SUBROUTINE FULOX

Handles read, modify, and admittance calls for fuel and lox

Commons EPARAM

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft ²)
AVGK(25)	REAL*4	average bulk modulus
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft ³)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
GF(25)	COMPLEX*8	admittance looking toward tank
IENG(25)	INTEGER*2	engine number
IGONE	INTEGER*2	flag for subroutine fuel or lox
ITANK(25)	INTEGER*2	tank number
ITLIN	INTEGER*2	flag indication fuel or lox
IUNIT	INTEGER*2	unit number of piping data file
IUNITP	INTEGER*2	unit number of working file
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft ²)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft ²)
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
MLINE	INTEGER*2	number of lines from tank
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PIPE1(75,25)	REAL*4	first parameter of pipe description
PIPE2(75,25)	REAL*4	second parameter of pipe description
PIPE3(75,25)	REAL*4	third parameter of pipe description
PIPE4(75,25)	REAL*4	fourth parameter of pipe description
PIPE5(75,25)	REAL*4	fifth parameter of pipe description
S	COMPLEX*8	current frequency
SECTN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT(25)	REAL*4	number of unique lines from pipe split
VOL(25)	REAL*4	volume of tank (ft ³)
VOLMF(25)	REAL*4	volume of manifold (ft ³)

Local Variables

ANS	CHAR*1	response to question
I	INTEGER*2	do loop index
IP	INTEGER*2	current segment number
IT	INTEGER*2	current tank number
QUEST3(2)	CHAR*40	question array
TTTL	CHAR*20	title from data file

COMPLEX FUNCTION FU1
Evaluates U1

Commons	CMPVAL	INTVAL	RELVAL
		Variables in Argument List	
XL		REAL*4	length of chamber
		Local Variables	
DX		REAL*4	integration increment
I		INTEGER*2	do loop index
VINT		COMPLEX*8	intermediate variable
X		REAL*4	current x location

SUBROUTINE GINERT
Evaluates curve fit of inertance of bends

		Variables in Argument List	
BEND		REAL*4	angle of bend (degrees)
X		REAL*4	ratio of inner to outer radius
Y		REAL*4	inertance
		Local Variables	
A		REAL*4	intermediate variable
B(3)		REAL*4	coefficient array for inertance fit

SUBROUTINE ITER
Iterates for dependent variable

Commons	CMPVAL	INTVAL	RELVAL	RESULT
		Variables in Argument List		
ID		INTEGER*2	flag for dependent variable	
TOL		REAL*4	convergence criteria	
		Local Variables		
FUN		REAL*4	intermediate variable	
FUN1		REAL*4	intermediate variable	
FUN2		REAL*4	intermediate variable	
I		INTEGER*2	do loop index	
VAL		REAL*4	intermediate variable	
VAL1		REAL*4	intermediate variable	
VAL2		REAL*4	intermediate variable	

SUBROUTINE LOX
Handles lox piping logic

Commons	EPARAM	FOPIPE	OPARAM	WCAOUT
		Variables in Argument List		
GOX(25)		COMPLEX*8	admittance looking toward tank	
IGONE		INTEGER*2	flag for subroutine fuel or lox	
IUNIT		INTEGER*2	unit number of lox data file	
IUNITP		INTEGER*2	unit number of lox work file	
S		COMPLEX*8	current frequency	
		Local Variables		

ANS	CHAR*1	response to question
LOXIN	CHAR*24	name of lox data file

SUBROUTINE NONDIM

Nondimensionalizes variables

Commons CMPVAL DIMVAL INTVAL PIPES RELVAL TTTL

Variables in Argument List

HOLD(20)	REAL*4	array for transferring variables
----------	--------	----------------------------------

K	INTEGER*2	engine number
---	-----------	---------------

Local Variables

CAREA	REAL*4	area of chamber
CSTARD	REAL*4	intermediate variable
FAC	REAL*4	intermediate variable
GC	REAL*4	gravitational constant (lbm-ft/lbf-sec ²)
I	INTEGER*2	do loop index
PEXTT	REAL*4	exit pressure
PI	REAL*4	mathematical constant
RFAR	REAL*4	intermediate variable
RHOBAR(50)	REAL*4	intermediate variable array
TAREA	REAL*4	throat area
UBARD(50)	REAL*4	intermediate variable array
VAR(13)	CHAR*8	names of nondimensional variables
VARD(20)	CHAR*8	names of dimensional variables

SUBROUTINE PLTALL

Plots n vs τ for all frequencies

Commons SFACT TTTL

Variables in Argument List

FREQ(NOF)	REAL*4	frequency array
LABLX	CHAR*8	label for x axis
LABLY	CHAR*8	label for y axis
M	INTEGER*2	number of frequencies
N	INTEGER*2	number of tau's
NENG	INTEGER*2	engine number
NOF	INTEGER*2	maximum number of frequencies
NOT	INTEGER*2	maximum number of tau's
X(NOT)	REAL*4	tau array
Y(NOT,NOF)	REAL*4	n array

Local Variables

ASPECT	REAL*4	intermediate variable
ETITL	CHAR*13	plot title
FREQI	CHAR*16	label for frequency
I	INTEGER*2	do loop index
IBOARD	INTEGER*2	flag for type of graphics board used
ICOLR	INTEGER*2	color flag
IEXTEN	INTEGER*2	extension of key hit
IFIL	INTEGER*2	color flag
IKEY	INTEGER*2	code of key hit
ILIN	INTEGER*2	color flag

IOPT	INTEGER*2	intermediate variable
IXLAB	INTEGER*2	intermediate variable
IXPIX	INTEGER*2	intermediate variable
IYLAB	INTEGER*2	intermediate variable
IYPIX	INTEGER*2	intermediate variable
J	INTEGER*2	do loop index
JCOL1	INTEGER*2	starting plot column
JCOL2	INTEGER*2	ending plot column
JROW1	INTEGER*2	starting plot row
JROW2	INTEGER*2	ending plot row
LABFAC(7)	CHAR*8	labels
MODE	INTEGER*2	graphics mode
MODET	INTEGER*2	text mode
NCOLT	INTEGER*2	number of text columns
RADHER(2)	CHAR*8	labels
XFAC	REAL*4	intermediate variable
XLABL(2)	CHAR*8	label
XMAJC	REAL*4	intermediate variable
XMAX	REAL*4	maximum x value for plot
XMIN	REAL*4	minimum x value for plot
XORG	REAL*4	plot x origin
YFAC	REAL*4	intermediate variable
YLABL(2)	CHAR*8	label
YMAJ	REAL*4	intermediate variable
YMAX	REAL*4	maximum y value for plot
YMIN	REAL*4	minimum y value for plot
YORG	REAL*4	plot y origin
YOVERX	REAL*4	intermediate variable

SUBROUTINE PLTVAR

Plots n vs τ for a single frequency

Commons	SFACT	TITL
Variables in Argument List		
FREQ	REAL*4	frequency
LABLX	CHAR*8	label for x axis
LABLY	CHAR*8	label for y axis
N	INTEGER*2	number of tau's
NENG	INTEGER*2	engine number
X(N)	REAL*4	tau array
Y(N)	REAL*4	n array
Local Variables		
ASPECT	REAL*4	intermediate variable
FREQI	CHAR*29	label for frequency
I	INTEGER*2	do loop index
IBOARD	INTEGER*2	flag for type of graphics board used
ICOLR	INTEGER*2	color flag
IEXTEN	INTEGER*2	extension of key hit
IFIL	INTEGER*2	color flag
IKEY	INTEGER*2	code of key hit
ILIN	INTEGER*2	color flag
IOPT	INTEGER*2	intermediate variable

IXLAB	INTEGER*2	intermediate variable
IYLAB	INTEGER*2	intermediate variable
JCOL1	INTEGER*2	starting plot column
JCOL2	INTEGER*2	ending plot column
JROW1	INTEGER*2	starting plot row
JROW2	INTEGER*2	ending plot row
LABFAC(7)	CHAR*8	labels
MODE	INTEGER*2	graphics mode
MODET	INTEGER*2	text mode
NCOLT	INTEGER*2	number of text columns
RADHER(2)	CHAR*8	labels
XFAC	REAL*4	intermediate variable
XLABL(2)	CHAR*8	label
XMAJ	REAL*4	intermediate variable
XMAX	REAL*4	maximum x value for plot
XMIN	REAL*4	minimum x value for plot
XORG	REAL*4	plot x origin
YFAC	REAL*4	intermediate variable
YLABL(2)	CHAR*8	label
YMAJ	REAL*4	intermediate variable
YMAX	REAL*4	maximum y value for plot
YMIN	REAL*4	minimum y value for plot
YORG	REAL*4	plot y origin
YOVERX	REAL*4	intermediate variable

SUBROUTINE READIN
Reads input data

Commons	CMPVAL	DIMVAL	EPARAM	INTVAL	RELVAL	TITL	WORK
Local Variables							
ANS		CHAR*1				response to question	
CDIAM		REAL*4				chamber diameter (ft)	
DCSDRD		REAL*4				d(cstar)/d(mixture ratio) (ft/sec)	
DHLDRD		REAL*4				d(enthalpy)/d(mixture ratio) (ft/sec) ²	
DTAUD		REAL*4				delta time lag (sec)	
GAMMAD		REAL*4				ratio of specific heats	
HOLD(20)		REAL*4				equivalenced to dimensioned variables	
I		INTEGER*2				do loop index	
IDATA		INTEGER*2				data source flag	
IGO		INTEGER*2				path flag	
II		INTEGER*2				do loop index	
J		INTEGER*2				do loop index	
K		INTEGER*2				do loop index	
LAMDAD		REAL*4				real part of complex frequency	
MEARD		REAL*4				mean comb. response function (lbm/sec)	
MUD		REAL*4				imaginary part of complex frequency	
NAME		CHAR*8				name of input parameter	
ND		REAL*4				pressure interaction index	
NRD		REAL*4				mixture ratio interaction index	
PCHMB		REAL*4				chamber pressure (lbf/ft ²)	
POOD		REAL*4				maximum pressure at injection face	
RBARD		REAL*4				mean mixture ratio	

RGAS	REAL*4	gas constant (ft ² /sec ² /°R)
RHOLOD	REAL*4	mass of liquid per unit chamber vol (lbm/ft ³)
TAUD	REAL*4	sensitive time lag (sec)
TCHMB	REAL*4	chamber temperature (°R)
TDIAM	REAL*4	throat diameter (ft)
ULOD	REAL*4	axial component of liquid velocity (ft/sec)
VALUE	REAL*4	value of input parameter
VAR(20)	CHAR*8	names of variables for printout
VARL(20)	CHAR*8	names of variables (lower case)
VARP(20)	CHAR*8	names of variables (upper case)
XLCD	REAL*4	x location of chamber-nozzle interface (ft)

SUBROUTINE RLINE

Reads fuel or lox file.

Commons EPARAM

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft ²)
AVGK(25)	REAL*4	average bulk modulus
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft ³)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
IENG(25)	INTEGER*2	engine number
ITANK(25)	INTEGER*2	tank number
IUNIT	INTEGER*2	unit number of fuel or lox file
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft ²)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft ²)
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe
LOOLD(25)	INTEGER*2	previous maximum number of iterations
MLINE	INTEGER*2	number of lines from tank
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PIPE1(75,25)	REAL*4	first parameter of pipe description
PIPE2(75,25)	REAL*4	second parameter of pipe description
PIPE3(75,25)	REAL*4	third parameter of pipe description
PIPE4(75,25)	REAL*4	fourth parameter of pipe description
PIPE5(75,25)	REAL*4	fifth parameter of pipe description
SECIN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT(25)	REAL*4	number of unique lines from pipe split
TITL	CHAR*20	title from input file
VOL(25)	REAL*4	volume of tank (ft ³)
VOLMF(25)	REAL*4	volume of manifold (ft ³)

Local Variables

ANS	CHAR*1	response to question
DIVAVG	REAL*4	intermediate variable
I	INTEGER*2	do loop index

IE	INTEGER*2	current engine number
IT	INTEGER*2	current tank number
J	INTEGER*2	do loop index
M	INTEGER*2	pointer
MM	INTEGER*2	do loop index
MTANK	INTEGER*2	number of tanks

SUBROUTINE RTYPE

Stores values for different types of piping

Variables in Argument List

AREA	REAL*4	area of pipe section (ft ²)
AVGK	REAL*4	average bulk modulus
CMAN	REAL*4	manifold capacitance
DENS	REAL*4	density of fluid (lbm/ft ³)
DIA	REAL*4	diameter of pipe section (ft)
KMAN	REAL*4	bulk modulus of manifold (lbf/ft ²)
L	REAL*4	length of pipe section (ft)
PCAP	REAL*4	capacitance of pipe section
PIND	REAL*4	inductance of pipe section
PIPE1	REAL*4	first parameter of pipe description
PIPE2	REAL*4	second parameter of pipe description
PIPE3	REAL*4	third parameter of pipe description
PIPE4	REAL*4	fourth parameter of pipe description
PIPE5	REAL*4	fifth parameter of pipe description
SECTN	INTEGER*2	pipe section type
VOLMF	REAL*4	volume of manifold (ft ³)

Local Variables

AREAB	REAL*4	area of pipe
DIME	REAL*4	diameter of pipe
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec ²)
PI	REAL*4	mathematical constant
VALUE	REAL*4	length of pipe

SUBROUTINE SETVAL

Sets value from iterated variable

Commons DIMVAL

Variables in Argument List

ID	INTEGER*2	pointer to variable
VAL	REAL*4	value of variable

SUBROUTINE SETVAR

Sets iterated variable from value

Commons CMPVAL DIMVAL INTVAL RELVAL

Variables in Argument List

ID	INTEGER*2	pointer to variable
VAL	REAL*4	value of variable

Local Variables		
ASTAR	REAL*4	speed of sound at injector face
PI	REAL*4	mathematical constant

SUBROUTINE TANKNO
Reads tank parameters

Variables in Argument List		
A(25)	REAL*4	speed of sound in the fluid (ft/sec)
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft ³)
IUNIT	INTEGER*2	unit number of fuel or lox file
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft ²)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
MTANK	INTEGER*2	number of tanks
VOL(25)	REAL*4	volume of tank (ft ³)

Local Variables		
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec ²)
I	INTEGER*2	do loop index

SUBROUTINE ZREAD
Reads input for input modification

Variables in Argument List		
NAME(8)	CHAR*1	name of input variable
VALUE	REAL*4	value of input variable

Local Variables		
BLK	CHAR*1	' '
CARD(80)	CHAR*1	card image
CEND(3)	CHAR*1	'E','N','D'
COMMA	CHAR*1	','
DCARD	CHAR*80	card image
E	CHAR*1	'E'
FRACT	REAL*4	fractional part of number
I	INTEGER*2	do loop index
ICOUNT	INTEGER*2	position counter
ID	INTEGER*2	position counter
II	INTEGER*2	position counter
J	INTEGER*2	do loop index
JJ	INTEGER*2	position counter
LE	CHAR*1	'e'
LEND(3)	CHAR*1	'e','n','d'
MINUS	CHAR*1	'-'
NUMBER(10)	CHAR*1	'0','1','2','3','4','5','6','7','8','9'
PERIOD	CHAR*1	'.'
PLUS	CHAR*1	'+'
POUND	CHAR*1	'#'
QUEST	CHAR*1	'?'
SIGN	REAL*4	sign of number or exponent
WHOLE	REAL*4	whole part of number

7.0 Program Listing

```
C
C      PROGRAM SSFREQ  03-24-92
C
C      Intermediate Mode Oscillations
C      Modified for n vs tau plots
C
C      This program will handle the following type elements
C
C      Straight pipes
C      Bends
C      Inline accumulators
C      Tuned stub accumulators
C      Helmholtz resonators
C      Parallel resonators
C      Pumps
C      Split pipes
C      Multiple tanks
C      Multiple engines
C
$LARGE
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX PP,UP,SIGP,FUNB,CVAR(17)
COMPLEX GFA(25),GOXA(25)
CHARACTER*1 ANS
CHARACTER*2 AM,PM,AP
CHARACTER*8 VARP(3),RADHER(2)
CHARACTER*24 ROCIN,ROCVAR,NAMENG
CHARACTER*40 TITLF
CHARACTER*60 TITLE
INTEGER*2 IHR,IMIN,ISEC,I100,IYR,IMON,IDAY
REAL FREQ(50),TAULST(50),MBAR,N,NR,LAMDA,MU,RVAR(13),NA(25)
COMMON /WORK/YP(50,50),YPA(50,50)
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*      S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*      DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /RESULT/PP,UP,SIGP,FUNB
COMMON /INTVAL/NVAL
COMMON /DIMVAL/HOLDD(20),XBARD(50),PBAR(50),TBAR(50)
COMMON /TITL/TITLE,TITLF,IHR,IMIN,AP,IYR,IMON,IDAY
COMMON /SFACT/SFAC
COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
EQUIVALENCE (N,RVAR(1)),(X1,CVAR(1))
DATA RADHER/' rad/sec',' Hertz '/'
DATA AM/'AM'/,PM/'PM'/
DATA VARP/' n ','tau(sec)',' MU '/'
DATA TOL/.0001/
DATA NOT/50/,NOF/50/
DATA IFREQ/0/,ITAU/0/
1 FORMAT(/' Eng. No.',3X,A8,5X,A8,5X,' FUNB(R)',5X,' FUNB(I)'/)
```



```

2 FORMAT(3X,I2,4X,1P4E13.5)
3 FORMAT(/' FREQUENCY =' ,1PE13.5,A)
  CALL GETTIM(IHR,IMIN,ISEC,I100)
  CALL GETDAT(IYR,IMON,IDAY)
  IYR=IYR-1900
  IF(IHR.LT.12) THEN
    AP=AM
  ELSE
    AP=PM
    IF(IHR.GT.12) IHR=IHR-12
  ENDIF
  CALL QCLEAR(0,7)
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'| Welcome to SSFREQ - an Intermediate Mode Program
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'| To send a plot to the printer
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'| The computer MUST be in GRAPHICS mode
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'| Hit PrScn to send the current plot to the printer
  WRITE(*,'(10X,A)')
*'|
  WRITE(*,'(10X,A)')
*'|
  SFAC=1.0
  WRITE(*,*) ' '
  WRITE(*,'(A)') ' If you want frequency in rad/sec, hit enter.'
  WRITE(*,'(A\)\') ' If you want it in Hertz, enter "H". '
  READ(*,'(A)')ANS
  IF(ANS.EQ.'H'.OR.ANS.EQ.'h') SFAC=6.283185
  OPEN(17,FORM='UNFORMATTED')
  OPEN(20,FORM='UNFORMATTED')
  WRITE(*,'(A\)\') ' Is the engine data on file ENG.RLN? (Y/N) '
  READ(*,'(A)')ANS
  IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
    NAMENG='ENG.RLN'
  ELSE
    WRITE(*,'(A\)\') ' Enter name of file with the engine data '
    READ(*,'(A)')NAMENG
  ENDIF
  OPEN(UNIT=9,FILE=NAMENG)
  JUNIT=9
  CALL ENGNO(JUNIT)

```

```

IGONE=2
CALL FUEL(S,GFA,11,16,IGONE)
IGONE=2
CALL LOX(S,GOXA,10,15,IGONE)
IGONE=0
WRITE(*,*)' '
WRITE(*,'(A\)' )' Are you are using IMODE.RLN for input data? '
READ(*,'(A)' )ANS
IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
  ROCIN='IMODE.RLN'
ELSE
  WRITE(*,'(A\)' )' Enter name of file containing input '
  READ(*,'(A)' )ROCIN
ENDIF
OPEN(12,FILE=ROCIN)
OPEN(14,FILE='IMODE.OUT')
XLC=1.0
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)'
                               Welcome to SSFREQ'
WRITE(*,*)' '
WRITE(*,*)'           Intermediate Mode Rocket Stability Aide'
WRITE(*,*)' '
WRITE(*,*)' There are three types of input, rocket parameters,'
WRITE(*,*)' Oxidizer feed parameters, and fuel feed parameters,'
WRITE(*,*)' Each may be read from files or from the keyboard'
WRITE(*,*)' '
WRITE(*,*)'           File Name                               Input'
WRITE(*,*)' '
WRITE(*,*)' IMODE.RLN or NAME read in   Rocket Parameters '
WRITE(*,*)' LOX.RLN                               Oxidizer Parameters'
WRITE(*,*)' FUEL.RLN                               Fuel Parameters  '
WRITE(*,*)' '
WRITE(*,*)' If keyboard entry, you will be prompted for values'
GO TO 22
21 CONTINUE
WRITE(*,*)' '
WRITE(*,'(A\)' )' Do you want to run another case? Enter Y or N '
READ(*,'(A)' )ANS
IF(ANS.EQ.'N'.OR.ANS.EQ.'n') STOP
WRITE(*,'(A\)' )' Do you wish to rewind engine file? '
READ(*,'(A)' )ANS
IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') REWIND JUNIT
CALL ENGNO(JUNIT)
IGONE=1
CALL FUEL(S,GFA,11,16,IGONE)
IGONE=1
CALL LOX(S,GOXA,10,15,IGONE)
IGONE=0

```

```

IF(IFREQ.NE.0) THEN
  WRITE(*,'(A\)' )' Do you wish to rewind frequency file? '
  READ(*,'(A\)' )ANS
  IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') REWIND 19
ENDIF
IF(ITAU.NE.0) THEN
  WRITE(*,'(A\)' )' Do you wish to rewind tau file? '
  READ(*,'(A\)' )ANS
  IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') REWIND 18
ENDIF
22 CONTINUE
CALL READIN
WRITE(*,*)' '
23 CONTINUE
WRITE(*,*)' Specify how frequency will be input -'
WRITE(*,*)' Enter R for a range of values'
WRITE(*,*)' Enter F for values in a file'
WRITE(*,*)' Enter K (end with -999) to enter values ',
* 'from keyboard'
READ(*,'(A\)' )ANS
IF(ANS.EQ.'R'.OR.ANS.EQ.'r') THEN
24 CONTINUE
  IF(SFAC.EQ.1.0) THEN
    WRITE(*,*)' Enter first and last values of frequency ',
    * 'in rad/sec and no. of points.'
  ELSE
    WRITE(*,*)' Enter first and last values of frequency ',
    * 'in hertz and no. of points.'
  ENDIF
  READ(*,*)STARTF,STOPF,NPTF
  IF(STARTF.EQ.0.0.AND.STOPF.EQ.0.0.AND.NPTF.EQ.0) GO TO 21
  IF(NPTF.GT.NO) THEN
    WRITE(*,*)' No. of points must be < ',NO
    GO TO 24
  ENDIF
  IF(STOPF.EQ.0.0) STOPF=STARTF
  IF(NPTF.EQ.0) NPTF=1
  IF(NPTF.EQ.1) THEN
    DELF=0.0
  ELSE
    DELF=(STOPF-STARTF)/(NPTF-1)
  ENDIF
  DO 25 I=1,NPTF
    FREQ(I)=STARTF+DELF*(I-1)
25 CONTINUE
  GO TO 27
ENDIF
IF(ANS.EQ.'F'.OR.ANS.EQ.'f') THEN
  IFREQ=1
  WRITE(*,*)' Is the frequency on IMODE.FRQ?'
  WRITE(*,'(A\)' )' Enter Y or N '
  READ(*,'(A\)' )ANS
  IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN

```

```

        OPEN(19,FILE='IMODE.FRQ')
    ELSE
        WRITE(*,'(A\)' )' Enter name of file for frequency '
        READ(*,'(A)' )ROCVAR
        OPEN(19,FILE=ROCVAR)
    ENDIF
    READ(19,*)NPTF
    IF(NPTF.GT.NO) THEN
        WRITE(*,*)' Too many points for program'
        GO TO 23
    ENDIF
    READ(19,*) (FREQ(I),I=1,NPTF)
    GO TO 27
ENDIF
IF(ANS.EQ.'K'.OR.ANS.EQ.'k') THEN
    NPTF=0
26 CONTINUE
    READ(*,*)VAR1
    IF(VAR1.EQ.-999) GO TO 27
    NPTF=NPTF+1
    FREQ(NPTF)=VAR1
    IF(NPTF.EQ.NO) GO TO 27
    GO TO 26
ELSE
    WRITE(*,*)' R, F, or K not entered, try again!'
    GO TO 23
ENDIF
27 CONTINUE
    WRITE(*,*)' Specify how tau will be input -'
    WRITE(*,*)' Enter R for a range of values'
    WRITE(*,*)' Enter F for values in a file'
    WRITE(*,*)' Enter K to enter values from keyboard'
    READ(*,'(A)' )ANS
    IF(ANS.EQ.'R'.OR.ANS.EQ.'r') GO TO 28
    IF(ANS.EQ.'F'.OR.ANS.EQ.'f') GO TO 30
    IF(ANS.EQ.'K'.OR.ANS.EQ.'k') GO TO 31
    WRITE(*,*)' R, F, or K not entered, try again!'
    GO TO 27
28 CONTINUE
    WRITE(*,*)' Enter first and last values of tau ',
    * 'and no. of points.'
    READ(*,*)STARTV,STOPV,NPTS
    IF(NPTS.GT.NO) THEN
        WRITE(*,*)' No. of points must be <',NO
        GO TO 28
    ENDIF
    IF(STOPV.EQ.0.0) STOPV=STARTV
    IF(NPTS.EQ.0) NPTS=1
    IF(NPTS.EQ.1) THEN
        DELVAL=0.0
    ELSE
        DELVAL=(STOPV-STARTV)/(NPTS-1)
    ENDIF

```

```

DO 29 I=1,NPTS
  TAULST(I)=STARTV+(I-1)*DELVAL
29 CONTINUE
  GO TO 33
30 CONTINUE
  ITAU=1
  WRITE(*,*) ' Is tau on IMODE.TAU?'
  WRITE(*, '(A\)' ) '          Enter Y or N '
  READ(*, '(A)' )ANS
  IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
    OPEN(18,FILE='IMODE.TAU')
  ELSE
    WRITE(*, '(A\)' ) ' Enter name of file for tau '
    READ(*, '(A)' )ROCVAR
    OPEN(18,FILE=ROCVAR)
  ENDIF
  READ(18,*)NPTS
  IF(NPTS.GT.NOT) THEN
    WRITE(*,*) ' Too many points for program'
    GO TO 27
  ENDIF
  READ(18,*) (TAULST(I),I=1,NPTS)
  GO TO 33
31 CONTINUE
  NPTS=0
32 CONTINUE
  WRITE(*, '(A\)' )
  * ' Enter new value for tau (-999 to stop) '
  READ(*,*,END=99)VAR1
  IF(VAR1.EQ.-999.0) GO TO 33
  NPTS=NPTS+1
  TAULST(I)=VAR1
  IF(NPTS.EQ.NOT) GO TO 33
  GO TO 32
33 CONTINUE
  DO 34 K=1,MENG
    NA(K)=1.0
34 CONTINUE
  REWIND 17
  DO 38 J=1,NPTF
    WRITE(14, '(1X,A)' )TITLE
    IF(SFAC.EQ.1.0) THEN
      WRITE(14,3)FREQ(J),RADHER(1)
      WRITE(*,3)FREQ(J),RADHER(1)
    ELSE
      WRITE(14,3)FREQ(J),RADHER(2)
      WRITE(*,3)FREQ(J),RADHER(2)
    ENDIF
    WRITE(14,1)VARP(2),VARP(1)
    WRITE(*,1)VARP(2),VARP(1)
    VARF=SFAC*FREQ(J)
    CALL SETVAR(VARF,6)
    CALL FUEL(S,GFA,11,16,IGONE)

```

```

CALL LOX(S,GOXA,10,15,IGONE)
DO 36 I=1,NPTS
  VART=TAULST(I)
  REWIND 20
  DO 35 K=1,MENG
    GF=GFA(K)
    GOX=GOXA(K)
    READ(20)HOLDD,XBARD,PBAR,TBAR
    CALL SETVAR(VARF,6)
    CALL SETVAR(VART,2)
    CALL SETVAR(NA(K),1)
    CALL ITER(1,TOL)
    NA(K)=HOLDD(1)
    YP(I,K)=HOLDD(1)
    WRITE(14,2)K,HOLDD(2),HOLDD(1),FUNB
    WRITE(*,2)K,HOLDD(2),HOLDD(1),FUNB
35  CONTINUE
36  CONTINUE
    WRITE(17)YP
    WRITE(*,'(A\)' )
    *  ' Do you wish to see n vs tau for this frequency? '
    READ(*,'(A)' )ANS
    IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
      IF(MENG.EQ.1) THEN
        CALL PLTVAR(TAULST,YP(1,1),NPTS,VARP(2),VARP(1),FREQ(J),1)
      ELSE
37  CONTINUE
        WRITE(*,'(A)' )' Specify which engine you wish to view'
        WRITE(*,'(A,I3,A\)' )' Enter 1 -',MENG,' or 0 to continue '
        READ(*,*)K
        IF(K.LT.0.OR.K.GT.MENG) THEN
          WRITE(*,*)' Invalid engine number, try again!'
          GO TO 37
        ELSEIF(K.NE.0) THEN
          CALL PLTVAR(TAULST,YP(1,K),NPTS,VARP(2),VARP(1),FREQ(J),K)
          GO TO 37
        ENDIF
      ENDIF
    ENDIF
38 CONTINUE
    DO 41 K=1,MENG
      REWIND 17
      DO 40 J=1,NPTF
        READ(17)YPA
        DO 39 I=1,NPTS
          YP(I,J)=YPA(I,K)
39  CONTINUE
40  CONTINUE
        CALL PLTALL(TAULST,YP,NOT,NOF,NPTS,NPTF,VARP(2),VARP(1),FREQ,K)
41 CONTINUE
      GO TO 21
99 CONTINUE
    STOP

```

```

END
SUBROUTINE ADMIT(S,GADM,A,AREA,CMAN,CTANK,DPROR,L,LFLOW,PMRAT,
*          SEGMN,SECTN,SPLIT,LOPEND,PCAP,PIND,IENG,TFLOW,
*          NOLINE,IP,ITLIN)
C      Determines admittance looking toward tank
CHARACTER*40 TITLE
CHARACTER*20 TITLF
INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
CHARACTER*2 AP
COMMON /WCATIT/TITLE,TITLF,IHR,IMIN,AP,IYR,IMON,IDAY
INTEGER SEGMN(25),SECTN(75,25)
INTEGER IENG(25),NOLINE(25)
REAL AREA(75,25),PCAP(75,25),PIND(75,25),L(75,25),LFLOW,ZO(75,25),
*      CMAN(25),DPROR(25),PMRAT(25),ZOR(25),TFLOW(25)
COMPLEX G(0:75,25),ZT(0:75,25),ZG(0:75,25),GOLD(0:75,25),GADM(25),
*      S,ZGEFF,ZTEFF
COMMON /WORK1/G,ZT,ZG
COMMON /WORK2/ZO
COMMON /SFACT/SFAC
COMPLEX CTANH,RHS,CFAC,CAPN,CAPM
CHARACTER*13 TYPEL(2)
DATA TYPEL/' in FUEL line',' in LOX line'/
DATA GRAV/32.2/
DATA IOPEN/0/
ZTOP=A/GRAV
TMASS=0.0
TCOUNT=0.0
DO 22 J=IP,IP+SPLIT
  GOLD(0,J)=0.0
  SECTN(SEGMN(J)+1,J)=0
  DO 21 I=1,SEGMN(J)
    GOLD(I,J)=0.0
    ZO(I,J)=0.0
    IF(SECTN(I,J).LE.2) THEN
      ZO(I,J)=ZTOP/AREA(I,J)
    ELSEIF(SECTN(I,J).EQ.7) THEN
      ZO(I,J)=0.0
    ELSE
      ZO(I,J)=SQRT(PIND(I,J)/PCAP(I,J))
    ENDIF
  21 CONTINUE
  IF(IENG(J).NE.0) THEN
    IE=IENG(J)
    ZOR(J)=2.0*DPROR(IE)/LFLOW
    IF(J.EQ.IP.AND.SPLIT.EQ.0.0) THEN
      TMASS=TFLOW(IE)
    ELSEIF(J.NE.IP) THEN
      TMASS=TMASS+NOLINE(J)*TFLOW(IE)
      TCOUNT=TCOUNT+NOLINE(J)
    ENDIF
  22 CONTINUE
  IF(TCOUNT.EQ.0.0) TCOUNT=1.0

```

```

G(0,IP)=CTANK*S
G(0,IP)=G(0,IP)/TCOUNT
ZT(0,IP)=1.0/G(0,IP)
DO 31 KLOOP=1,LOPEND
  DO 25 J=IP,IP+SPLIT
    IF(J.NE.IP) THEN
      G(0,J)=G(SEGMN(IP),IP)
      ZT(0,J)=1.0/G(0,J)
    ENDIF
  DO 24 I=1,SEGMN(J)
    ZGEFF=G(I-1,J)
    IF(SECTN(I,J).LE.1) THEN
C      bend in pipe or straight section
      TL=L(I,J)/A
      IF(KLOOP.NE.1.AND.SPLIT.NE.0.AND.J.NE.IP.AND.I.EQ.1) THEN
        ZGEFF=0.0
        DO 23 K=IP+1,IP+SPLIT
          IE=IENG(K)
          IF(K.EQ.J) THEN
            ZGEFF=ZGEFF+(NOLINE(K)-1.0)/ZG(0,K)
          ELSE
            ZGEFF=ZGEFF+NOLINE(K)/ZG(0,K)
          ENDIF
        CONTINUE
      ZGEFF=G(SEGMN(IP),IP)+ZGEFF
    ENDIF
    G(I,J)=(1.0+CTANH(S*TL)/(ZGEFF*ZO(I,J)))/(1.0+ZGEFF*
*      ZO(I,J)*CTANH(S*TL))
    ELSEIF(SECTN(I,J).EQ.2) THEN
C      inline resonator
      G(I,J)=1.0+PCAP(I,J)*S/ZGEFF
    ELSEIF(SECTN(I,J).EQ.3) THEN
C      tuned stub
      G(I,J)=1.0+CTANH(S*SQRT(PIND(I,J)*PCAP(I,J)))/(ZO(I,J)*
*      ZGEFF)
    ELSEIF(SECTN(I,J).EQ.4) THEN
C      helmholtz resonator
      G(I,J)=1.0+S*PCAP(I,J)/(1.0+PIND(I,J)*PCAP(I,J)*S**2)/ZGEFF
    ELSEIF(SECTN(I,J).EQ.5) THEN
C      parallel resonator
      G(I,J)=PIND(I,J)*PCAP(I,J)*S**2+1.0
      G(I,J)=G(I,J)/(G(I,J)+PIND(I,J)*S*ZGEFF)
    ELSEIF(SECTN(I,J).EQ.6) THEN
C      pump
      G(I,J)=(1.0+PCAP(I,J)*S/ZGEFF)/(1.0+(PIND(I,J)*S+
*      AREA(I,J))*(PCAP(I,J)*S+ZGEFF))
    ELSEIF(SECTN(I,J).EQ.7) THEN
      G(SEGMN(J),J)=1.0+CMAN(J)*S/ZGEFF
    ENDIF
    G(I,J)=G(I,J)*ZGEFF
    ZT(I,J)=1.0/G(I,J)
  24 CONTINUE
  IF(SPLIT.NE.0.0.AND.J.EQ.IP) GO TO 25

```



```

      G(SEGMN(J)+1,J)=1.0/(1.0+ZOR(J)*G(SEGMN(J),J))
      G(SEGMN(J)+1,J)=G(SEGMN(J)+1,J)*G(SEGMN(J),J)
25  CONTINUE
      IF(LOPEND.EQ.1.OR.SPLIT.EQ.0.0) GO TO 31
      DO 28 J=IP+SPLIT,IP,-1
        IF(J.EQ.IP) THEN
          LOPHI=SEGMN(J)
        ELSE
          ZG(SEGMN(J)-1,J)=ZOR(J)/(ZOR(J)*CMAN(J)*S+1.0)
          LOPHI=SEGMN(J)-2
        I=LOPHI+1
      ENDIF
      IF(LOPHI.NE.0) THEN
        DO 27 I=LOPHI,1,-1
          IF(I.EQ.LOPHI.AND.J.EQ.IP) THEN
            ZG(I,J)=0.0
            ZTEFF=ZT(I-1,J)
            DO 26 K=IP+1,IP+SPLIT
              ZGEFF=ZG(1,K)
              ZOEFF=ZO(1,K)
              ZLP=L(1,K)
              TL=(L(I,J)+ZLP)/A
              CAPN=(ZOEFF-ZTEFF)/(ZOEFF+ZTEFF)
              CAPM=(ZOEFF-ZGEFF)/(ZOEFF+ZGEFF)
              CFAC=CEXP(-2.0*S*TL)
              RHS=(ZOEFF+ZGEFF)*(1.0-CAPN*CAPM*CFAC)*CEXP(S*ZLP/A)
              CFAC=CAPN*CFAC*CEXP(2.0*S*ZLP/A)
              ZG(0,K)=(RHS-ZOEFF*(1.0-CFAC))/(1.0+CFAC)
              ZG(I,J)=ZG(I,J)+NOLINE(K)/ZG(0,K)
            26  CONTINUE
            ZG(I,J)=1.0/ZG(I,J)
          ELSE
            ZGEFF=ZG(I+1,J)
            ZOEFF=ZO(I+1,J)
            ZLP=L(I+1,J)
            ZTEFF=ZT(I-1,J)
            IF(SECTN(I+1,J).LE.1) THEN
C          bend in pipe or straight section
              TL=(L(I,J)+ZLP)/A
              CAPN=(ZOEFF-ZTEFF)/(ZOEFF+ZTEFF)
              CAPM=(ZOEFF-ZGEFF)/(ZOEFF+ZGEFF)
              CFAC=CEXP(-2.0*S*TL)
              RHS=(ZOEFF+ZGEFF)*(1.0-CAPN*CAPM*CFAC)*CEXP(S*ZLP/A)
              CFAC=CAPN*CFAC*CEXP(2.0*S*ZLP/A)
              ZG(I,J)=(RHS-ZOEFF*(1.0-CFAC))/(1.0+CFAC)
            ELSEIF(SECTN(I+1,J).EQ.2) THEN
C          inline resonator
              ZG(I,J)=ZGEFF/(ZGEFF*PCAP(I+1,J)*S+1.0)
            ELSEIF(SECTN(I+1,J).EQ.3) THEN
C          tuned stub
              ZG(I,J)=ZOEFF/CTANH(S*SQRT(PIND(I+1,J)*PCAP(I+1,J)))
              ZG(I,J)=(ZG(I,J)*ZGEFF)/(ZG(I,J)+ZGEFF)
            ELSEIF(SECTN(I+1,J).EQ.4) THEN

```

```

C      helmholtz resonator
      ZG(I,J)=(1.0+PIND(I+1,J)*PCAP(I+1,J)*S**2)/(PCAP(I+1,J)*S)
      ZG(I,J)=(ZG(I,J)*ZGEFF)/(ZG(I,J)+ZGEFF)
      ELSEIF(SECTN(I+1,J).EQ.5) THEN
C      parallel resonator
      ZG(I,J)=ZGEFF+PIND(I+1,J)*S/(PIND(I+1,J)*PCAP(I+1,J)*S**2+
*      1.0)
      ELSEIF(SECTN(I+1,J).EQ.6) THEN
C      pump
      ZG(I,J)=ZGEFF+PIND(I+1,J)*S-AREA(I+1,J)
      ZG(I,J)=ZG(I,J)/(1.0+ZG(I,J)*PCAP(I+1,J)*S)
      ENDIF
      ENDIF
27  CONTINUE
      ENDIF
28  CONTINUE
      ERRP=0.0
      DO 30 J=IP,IP+SPLIT
      DO 29 I=1,SEGMN(J)
      GDIF=CABS(GOLD(I,J))
      IF(GDIF.NE.0.0) GDIF=ABS(GDIF-CABS(G(I,J)))/GDIF
      IF(GDIF.GT.ERRP) THEN
      ERRP=GDIF
      WG=CABS(G(I,J))
      WGOLD=CABS(GOLD(I,J))
      IWG=I
      JWG=J
      ENDIF
      GOLD(I,J)=G(I,J)
29  CONTINUE
30  CONTINUE
      IF(KLOOP.GT.1.AND.ERRP.LT.0.001) GO TO 32
31  CONTINUE
      IF(IOPEN.EQ.1) GO TO 32
      IF(IOPEN.EQ.0) THEN
      OPEN(UNIT=13,FILE='SURF.ERR')
      WRITE(13,*) ' '
      WRITE(13,*) ' '
      WRITE(13,*) TITLE
      WRITE(13,*) ' '
      IOPEN=1
      ENDIF
      WRITE(13, '(' ' jw =', F8.1, ' after', I3, ' iterations',
*      ' has error of', F8.3, '% ', A) ')
*      AIMAG(S)/SFAC, IOPEN, 100.0*ERRP, TYPEL(ITLIN)
      WRITE(13, '(10X, ' I=', I3, 3X, ' J=', I3, 3X, ' |G|=',
*      ' |GOLD|=', E12.4) ') IWG, JWG, WG, WGOLD
32  CONTINUE
      DO 35 J=IP,IP+SPLIT
      IF(IENTG(J).EQ.0) THEN
      RATPM=0.0
      DO 33 I=IP+1,IP+SPLIT
      RATPM=RATPM+PMRAT(IENG(I))*NOLINE(I)

```

```

33  CONTINUE
    RATPM=RATPM/TCOUNT
    LOPHI=SEGMN(J)
    ELSE
        RATPM=PMRAT(IENG(J))
        IF(NOLINE(J).NE.0) RATPM=RATPM*NOLINE(J)
        LOPHI=SEGMN(J)+1
    ENDIF
    DO 34 I=0,LOPHI
        G(I,J)=RATPM*G(I,J)
34  CONTINUE
    IF(IENG(J).EQ.0) GO TO 35
    IEE=IENG(J)
    GADM(IEE)=GADM(IEE)+G(LOPHI,J)
35  CONTINUE
    RETURN
    END
    SUBROUTINE BENDS(PIPE1,PIPE2,PIPE3,PIPE4,VALUE,DIME)
C    Computes effective straight pipe for bend
    REAL LBEND
    LBEND=0.0174533*PIPE1*ABS(PIPE2)
    RATIO=(PIPE1-0.5*PIPE3)/(PIPE1+0.5*PIPE3)
    CALL GINERT(ABS(PIPE2),RATIO,Y)
    GAMMA=(LBEND+Y*PIPE3)/LBEND
    VALUE=GAMMA*(LBEND+2.0*PIPE4)
    DIME=PIPE3/(GAMMA)**0.25
    RETURN
    END
    SUBROUTINE BOUND(PP,UP,SIGP,FUNB)
C    Evaluates the boundary function
    COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*           S,GF,GOX,RFA,RFC
    COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLD, CSTAR,
*           DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
    COMMON /INTVAL/NVAL
    REAL MBAR,N,NR,LAMDA,MU
    COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,GF,GOX,U1,RFH,RFK,RFP,RFA,RFC
    COMPLEX FP1,FU1,FSIGP,PP,UP,SIGP,FUNB,CSINH,COOSH
    P1=FP1(XLC)
    U1=FU1(XLC)
    P0=P00*COOSH(S*XLC)
    U0=-(1.0/GAMMA)*P00*CSINH(S*XLC)
    PP=P0+P1
    UP=U0+U1
    SIGP=FSIGP(XLC)
    FUNB=UP+RFA*PP+RFC*SIGP
    RETURN
    END
    COMPLEX FUNCTION COOSH(S)
C    Evaluates the complex hyperbolic cosine
    COMPLEX S
    REAL LAMDA, MU
    LAMDA=REAL(S)

```

```

MU=AIMAG(S)
COSHR=COSH(LAMDA)*COS(MU)
COSHI=SINH(LAMDA)*SIN(MU)
COOSH=CMPLX(COSHR,COSHI)
RETURN
END
C    COMPLEX FUNCTION CSINH(S)
      Evaluates the complex hyperbolic sine
COMPLEX S
REAL LAMDA, MU
LAMDA=REAL(S)
MU=AIMAG(S)
SINHR=SINH(LAMDA)*COS(MU)
SINHI=COSH(LAMDA)*SIN(MU)
CSINH=CMPLX(SINHR,SINHI)
RETURN
END
C    COMPLEX FUNCTION CTANH(S)
      Evaluates the complex hyperbolic tangent
COMPLEX S, CTANN, CTAND, CSINH, COOSH
CTANN=CSINH(S)
CTAND=COOSH(S)
CTANH=(0.0,0.0)
IF(CTAND.NE.0.0) CTANH=CTANN/CTAND
RETURN
END
C    SUBROUTINE ENGNO(IUNIT)
      Reads engine parameters
COMMON /EPARAM/MENG, TFLOW(25), PCHMB(25), DPROR(25), PMRAT(25)
READ(IUNIT,*) MENG
IF(MENG.GT.25) THEN
  WRITE(*,*) ' Number of engines must be less than 25'
  STOP
ENDIF
IF(MENG.LE.0) MENG=1
DO 21 I=1,MENG
  READ(IUNIT,*) TFLOW(I), PCHMB(I), DPROR(I)
  PMRAT(I)=PCHMB(I)/TFLOW(I)
21 CONTINUE
RETURN
END
C    SUBROUTINE EVAL(X)
      Evaluates parameters at a given x location
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*          S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*          DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
REAL MBAR,N,NR,LAMDA,MU
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX CSINH,COOSH
IF(NVAL.EQ.1) THEN
  UB=UBAR(1)

```

```

        GO TO 23
    ENDIF
    DO 21 I=2,NVAL
        IF(X.LE.XBAR(I)) GO TO 22
21  CONTINUE
        UB=UBAR(NVAL)
        GO TO 23
22  CONTINUE
        FAC=(X-XBAR(I-1))/(XBAR(I)-XBAR(I-1))
        UB=UBAR(I-1)+FAC*(UBAR(I)-UBAR(I-1))
23  CONTINUE
        RFH=(1.0+RBAR)*((RBAR/CSTAR)*DCSDR-NR*S*TAU)*(GOX
*      -RBAR*GF)/RBAR
        RFK=(1.0+S*TAUT)*(GOX+GF)
        RFP=N*(1.0-CEXP(S*TAU))
        P0=P00*CCOSH(S*X)
        U0=-(1.0/GAMMA)*P00*CSINH(S*X)
        X1=(GAMMA-1.0)*UB*U0+(1.0+RBAR)*DHLDR*(MBAR/S)
*      *CEXP(-S*TAUT)*(GOX-RBAR*GF)*P00
        Y1=-UB*P0
        Z1=(1.0/GAMMA)*UB*P0+RHOLO*ULO
        W1=2.0*UB*U0
        M1=MBAR*(CEXP(-S*TAUT)*(RFK+RFH)*P00-RFP*P0)
        RETURN
    END
    COMPLEX FUNCTION FP1(XL)
    C      Evaluates P1
    COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*      S,GF,GOX,RFA,RFC
    COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*      DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
    COMMON /INTVAL/NVAL
    REAL MBAR,N,NR,LAMDA,MU
    COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
    COMPLEX CSINH,CCOSH
    COMPLEX VINT
    DX=XL/50.0
    FP1=CMPLX(0.0,0.0)
    DO 21 I=1,51
        X=(I-1)*DX
        CALL EVAL(X)
        VINT=(S*(W1-X1)+M1)*CSINH(S*(XL-X))
*      +S*(Y1+Z1)*CCOSH(S*(XL-X))
        IF(I.EQ.1.OR.I.EQ.51) THEN
            FP1=FP1+0.5*VINT*DX
        ELSE
            FP1=FP1+VINT*DX
        ENDIF
21  CONTINUE
        FP1=-GAMMA*(W1+FP1)
        RETURN
    END
    COMPLEX FUNCTION FSIGP(XL)

```

```

C      Evaluates SIG'
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*      S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDLDR,CSTAR,
*      DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
REAL MBAR,N,NR,LAMDA,MU
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
REAL UB(51)
COMPLEX VINT(51),VVINT(51),FSIG2,FCON
DX=XL/50.0
DO 23 I=1,51
  X=(I-1)*DX
  IF(NVAL.EQ.1) THEN
    UB(I)=UBAR(1)
    GO TO 23
  ENDIF
  DO 21 II=2,NVAL
    IF(X.LE.XBAR(II)) GO TO 22
21 CONTINUE
    II=NVAL
22 CONTINUE
    FAC=(X-XBAR(II-1))/(XBAR(II)-XBAR(II-1))
    UB(I)=UBAR(II-1)+FAC*(UBAR(II)-UBAR(II-1))
23 CONTINUE
  DO 24 I=1,51
    X=(I-1)*DX
    CALL EVAL(X)
    VINT(I)=(GAMMA-1.0)/GAMMA*P0
    VVINT(I)=1.0/UB(I)
24 CONTINUE
    FCON=(1.0+RBAR)*DHLDLDR*(GOX-RBAR*GF)*P00
    *      *CEXP(-S*TAUT)
  DO 26 I=1,51
    FSIG2=CMPLX(0.0,0.0)
    DO 25 J=I,51
      IF(J.EQ.I.OR.J.EQ.51) THEN
        FSIG2=FSIG2+0.5*VVINT(J)*DX
      ELSE
        FSIG2=FSIG2+VVINT(J)*DX
      ENDIF
    DO 27 I=1,51
      IF(I.EQ.1.OR.I.EQ.51) THEN
        FSIGP=FSIGP+0.5*VINT(I)*DX
      ELSE
        FSIGP=FSIGP+VINT(I)*DX
      ENDIF
27 CONTINUE

```

```

FSIGP=-FSIGP/UB(51)
RETURN
END
SUBROUTINE FUEL(S,GF,IUNIT,IUNITP,IGONE)
C   Handles fuel piping logic
COMPLEX GF(25),S
COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
INTEGER SEGMN(25),SECTN(75,25),NOLINE(25),IENG(25),ITANK(25),
*   LOPOLD(25),LOPEND(25)
REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
COMMON /FPARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
*   DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
*   AREA(75,25),DIA(75,25),L,PIND(75,25),
*   PCAP(75,25),AVGK(25),
*   SEGMN,SECTN,NOLINE,IENG,ITANK,LOPOLD,LOPEND
COMMON /FOPIPE/PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),
*   PIPE4(75,25),PIPE5(75,25)
CHARACTER*24 FUELIN,NAMLIN(2)
COMMON /WCAOUT/NAMLIN,IUNITH
CHARACTER*1 ANS
IF(IGONE.EQ.2) THEN
  WRITE(*,'(A\)' )' Is fuel line data in a file? (Y/N) '
  READ(*,'(A)' )ANS
  IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
    WRITE(*,'(A\)' )' Is the file name FUEL.RLN? (Y/N) '
    READ(*,'(A)' )ANS
    IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
      OPEN(UNIT=IUNIT,FILE='FUEL.RLN')
      NAMLIN(1)='FUEL.RLN'
    ELSE
      WRITE(*,'(A\)' )' Enter name of file with fuel line data '
      READ(*,'(A)' )FUELIN
      OPEN(IUNIT,FILE=FUELIN)
      NAMLIN(1)=FUELIN
    ENDIF
  ENDIF
ENDIF
OPEN(IUNITP,FORM='UNFORMATTED')
ENDIF
IUNITH=IUNIT
CALL FULOX(S,GF,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,
*   A,AREA,AVGK,CMAN,CTANK,DENS,DIA,IENG,IGONE,ITANK,
*   IUNIT,IUNITP,KMAN,KTANK,L,LOPEND,LOPOLD,LFLOW,MLINE,NOLINE,PCAP,
*   PIND,SPLIT,VOL,VOLMF,1)
RETURN
END
SUBROUTINE FULOX(S,GF,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,
*   A,AREA,AVGK,CMAN,CTANK,DENS,DIA,IENG,IGONE,ITANK,
*   IUNIT,IUNITP,KMAN,KTANK,L,LOPEND,LOPOLD,LFLOW,MLINE,NOLINE,PCAP,
*   PIND,SPLIT,VOL,VOLMF,ITLIN)
C   Handles read, modify, and admittance calls for fuel and lox
COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
INTEGER SEGMN(25),SECTN(75,25),NOLINE(25),IENG(25),ITANK(25),
*   LOPOLD(25),LOPEND(25)

```

```

      REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
      REAL SPLIT(25),A(25),CMAN(25),CTANK(25),
*          DENS(25),VOL(25),VOLMF(25),
*          AREA(75,25),DIA(75,25),PIND(75,25),
*          PCAP(75,25),AVGK(25)
      REAL PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),
*          PIPE4(75,25),PIPE5(75,25)
      COMPLEX GF(25),S
      CHARACTER*20 TITL
      CHARACTER*1 ANS
      CHARACTER*40 QUEST3(2)
      DATA QUEST3/' Do you wish to rewind fuel line file? ',
*                ' Do you wish to rewind lox line file? '/
      IF(IGONE.EQ.2) THEN
        CALL RLINE(TITL,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,
* PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
* A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
* AVGK,MLINE)
        REWIND IUNITP
        WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
      ELSEIF(IGONE.EQ.0) THEN
        DO 21 I=1,MENG
          GF(I)=0.0
21      CONTINUE
          IP=1
          DO 22 I=1,MLINE
            IT=ITANK(I)
            CALL ADMIT(S,GF,A(IT),AREA,CMAN,CTANK(IT),DPROR,
* L,LFLOW(IT),PMRAT,SEGMN,SECTN,
* SPLIT(I),LOPEND(I),PCAP,PIND,IENG,TFLOW,
* NOLINE,IP,ITLIN)
            IP=IP+SPLIT(I)+1
22      CONTINUE
          RETURN
      ELSEIF(IGONE.EQ.1) THEN
        WRITE(*,'(A\')')QUEST3(ITLIN)
        READ(*,'(A)')ANS
        IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') REWIND IUNIT
        CALL RLINE(TITL,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,
* PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
* A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
* AVGK,MLINE)
        REWIND IUNITP
        WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
        IGONE=0
      ENDIF
      RETURN
      END
      COMPLEX FUNCTION FU1(XL)
C      Evaluates U1
      COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*          S,GF,GOX,RFA,RFC
      COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDLDR,CSTAR,

```



```

*          DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
REAL MBAR,N,NR,LAMDA,MU
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX CSINH,COOSH
COMPLEX VINT
DX=XL/50.0
FU1=CMPLX(0.0,0.0)
DO 21 I=1,51
  X=(I-1)*DX
  CALL EVAL(X)
  VINT=(S*(W1-X1)+M1)*COOSH(S*(XL-X))
*      +S*(Y1+Z1)*CSINH(S*(XL-X))
  IF(I.EQ.1.OR.I.EQ.51) THEN
    FU1=FU1+0.5*VINT*DX
  ELSE
    FU1=FU1+VINT*DX
  ENDIF
21 CONTINUE
FU1=Y1+FU1
RETURN
END
SUBROUTINE GINERT(BEND,X,Y)
C      Evaluates curve fit of inertance of bends
DIMENSION B(3)
DATA B/0.0,0.7877014E-02,-0.2814679E-04/
A=B(1)+(B(2)+B(3)*BEND)*BEND
Y=A*(X-1.0)**2
RETURN
END
SUBROUTINE ITER(ID,TOL)
C      Iterates for dependent variable
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*          S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*          DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
COMMON /RESULT/PP,UP,SIGP,FUNB
REAL MBAR,N,NR,LAMDA,MU,RVAR(13)
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX PP,UP,SIGP,FUNB,CVAR(17)
EQUIVALENCE (N,RVAR(1)),(X1,CVAR(1))
CALL SETVAL(VAL1,ID)
CALL BOUND(PP,UP,SIGP,FUNB)
FUN1=REAL(FUNB)
IF(ABS(FUN1).LE.TOL) GO TO 22
VAL2=1.01*VAL1
IF(VAL1.EQ.0) VAL2=0.01
CALL SETVAR(VAL2,ID)
CALL BOUND(PP,UP,SIGP,FUNB)
FUN2=REAL(FUNB)
IF(ABS(FUN2).LE.TOL) GO TO 22
IF(FUN1.EQ.FUN2) THEN

```

```

      VAL=VAL1+VAL2
    ELSE
      VAL=VAL1-FUN1*(VAL2-VAL1)/(FUN2-FUN1)
    ENDIF
    IF (ABS(FUN2).LT.ABS(FUN1)) THEN
      FUN=FUN2
      FUN2=FUN1
      FUN1=FUN
      VAL=VAL2
      VAL2=VAL1
      VAL1=VAL
    ENDIF
    DO 21 I=1,20
      CALL SETVAR(VAL,ID)
      CALL BOUND(PP,UP,SIGP,FUNB)
      FUN=REAL(FUNB)
      IF (ABS(FUN).LE.TOL) GO TO 22
      IF (ABS(FUN).LT.ABS(FUN1)) THEN
        FUN2=FUN1
        FUN1=FUN
        VAL2=VAL1
        VAL1=VAL
      ELSE
        FUN2=FUN
        VAL2=VAL
      ENDIF
      IF (FUN1.EQ.FUN2) THEN
        IF (VAL1.EQ.VAL2) THEN
          VAL=VAL1+VAL2
        ELSE
          VAL=0.5*(VAL1+VAL2)
        ENDIF
      ELSE
        VAL=VAL1-FUN1*(VAL2-VAL1)/(FUN2-FUN1)
      ENDIF
    21 CONTINUE
    WRITE(*,*) ' FAILED TO CONVERGE after 20 iterations'
    22 CONTINUE
    RETURN
  END
  SUBROUTINE LOX(S,GOX,IUNIT,IUNITP,IGONE)
C    Handles fuel piping logic
    COMPLEX GOX(25),S
    COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
    INTEGER SEGMN(25),SECTN(75,25),NOLINE(25),IENG(25),ITANK(25),
    *    LOPOLD(25),LOPEND(25)
    REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
    COMMON /OPARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
    *    DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
    *    AREA(75,25),DIA(75,25),L,PIND(75,25),
    *    PCAP(75,25),AVGK(25),
    *    SEGMN,SECTN,NOLINE,IENG,ITANK,LOPOLD,LOPEND
    COMMON /FOPIPE/PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),

```

```

*          PIPE4(75,25),PIPE5(75,25)
CHARACTER*24 LOXIN,NAMLIN(2)
COMMON /WCAOUT/NAMLIN,IUNITH
CHARACTER*1 ANS
IF(IGONE.EQ.2) THEN
  WRITE(*,'(A\\)') ' Is lox line data in a file? (Y/N) '
  READ(*,'(A)')ANS
  IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
    WRITE(*,'(A\\)') ' Is the file name LOX.RLN? (Y/N) '
    READ(*,'(A)')ANS
    IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
      OPEN(UNIT=IUNIT,FILE='LOX.RLN')
      NAMLIN(2)='LOX.RLN'
    ELSE
      WRITE(*,'(A\\)') ' Enter name of file with lox line data '
      READ(*,'(A)')LOXIN
      OPEN(IUNIT,FILE=LOXIN)
      NAMLIN(2)=LOXIN
    ENDIF
  ENDIF
  OPEN(IUNITP,FORM='UNFORMATTED')
ENDIF
IUNITH=IUNIT
CALL FULOX(S,GOX,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,
* A,AREA,AVGK,CMAN,CTANK,DENS,DIA,IENG,IGONE,ITANK,
* IUNIT,IUNITP,KMAN,KTANK,L,LOPEND,LOPOLD,LFLOW,MLINE,NOLINE,PCAP,
* PIND,SPLIT,VOL,VOLMF,2)
RETURN
END
SUBROUTINE NONDIM(HOLD,K)
C      Nondimensionalizes variables
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*      S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*      DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
COMMON /DIMVAL/HOLDD(20),XBARD(50),PBAR(50),TBAR(50)
COMMON /PIPES/PFACE,TFACE,ASTAR
INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
CHARACTER*2 AP
CHARACTER*60 TTITLE
CHARACTER*40 TTTLF
COMMON /TTTL/TTITLE,TTTLF,IHR,IMIN,AP,IYR,IMON,IDAY
REAL MBAR,N,NR,LAMDA,MU,RVAR(15)
REAL MBARD,ND,NRD,LAMDAD,MUD
REAL HOLD(20),UBARD(50),RHOBAR(50)
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX CVAR(17)
CHARACTER*8 VAR(13),VARD(20)
EQUIVALENCE (N,RVAR(1)),(X1,CVAR(1))
EQUIVALENCE
*      (ND,HOLDD(1)),(TAUD,HOLDD(2)),(DTAUD,HOLDD(3)),
*      (NRD,HOLDD(4)),(LAMDA,HOLDD(5)),(MUD,HOLDD(6)),

```

```

*      (CDIAM,HOLDD(7)),(TDIAM,HOLDD(8)),(XLCD,HOLDD(9)),
*      (GAMMAD,HOLDD(10)),(RGAS,HOLDD(11)),(P00D,HOLDD(12)),
*      (MBARD,HOLDD(13)),(RBARD,HOLDD(14)),(DCSDRD,HOLDD(15)),
*      (DHLDRD,HOLDD(16)),(RHOLOD,HOLDD(17)),(ULOD,HOLDD(18)),
*      (PCHMB,HOLDD(19)),(TCHMB,HOLDD(20))
DATA VAR/'      N=','      TAU=','      DTAU=','      NR=','      RBAR=','
*      '      MBAR=','      GAMMA=','      P00=','      DHLDR=','      CSTAR=','
*      '      DCSDR=','      RHOLO=','      ULO='/'
DATA VARD/'      N=','      TAU=','      DTAU=','      NR=','      LAMDA=','
*      '      MU=','      CDIAM=','      TDIAM=','      XLC=','      GAMMA=','
*      '      RGAS=','      P00=','      MBAR=','      RBAR=','      DCSDR=','
*      '      DHLDR=','      RHOLO=','      ULO=','      PCHMB=','      TCHMB='/'
DATA PI/3.141593/,GC/32.174/
1 FORMAT(A8,1PE13.5,2X,A8,E13.5,2X,A8,E13.5)
2 FORMAT(' ')
DO 21 I=1,20
    HOLDD(I)=HOLD(I)
21 CONTINUE
    IF(PCHMB.NE.PBAR(1)) THEN
        FAC=PCHMB/PBAR(1)
        DO 22 I=1,NVAL
            PBAR(I)=FAC*PBAR(I)
22 CONTINUE
        ENDIF
        IF(TCHMB.NE.TBAR(1)) THEN
            FAC=TCHMB/TBAR(1)
            DO 23 I=1,NVAL
                TBAR(I)=FAC*TBAR(I)
23 CONTINUE
            ENDIF
            CAREA=0.25*PI*CDIAM**2
            WRITE(14,2)
            WRITE(14,*) ' CAREA=',CAREA
            TAREA=0.25*PI*TDIAM**2
            WRITE(14,*) ' TAREA=',TAREA
            PFACE=PBAR(1)
            PEXIT=PBAR(NVAL)
            TFACE=MBARD
            ASTAR=SQRT(GAMMAD*RGAS*TBAR(1))
            WRITE(14,*) ' ASTAR=',ASTAR
            CSTARD=PEXIT*TAREA*GC/MBARD
            WRITE(14,*) ' CSTARD=',CSTARD
            DO 24 I=1,NVAL
                RHOBAR(I)=PBAR(I)*GC/(RGAS*TBAR(I))
                WRITE(14,*) ' RHOBAR=',RHOBAR(I)
                UBARD(I)=MBARD/(RHOBAR(I)*CAREA)
                WRITE(14,*) ' UBARD=',UBARD(I)
24 CONTINUE
                N=ND
                TAU=TAUD*ASTAR/XLCD
                DTAU=DTAUD*ASTAR/XLCD
                TAUT=TAU+DTAU
                NR=NRD

```

```

RBAR=RBARD
MBAR=MBARD/(RHOBAR(1)*ASTAR*CAREA/XLCD)
GAMMA=GAMMAD
POO=POOD/PBAR(1)
DHLDR=DHLDRD
CSTAR=CSTARD/ASTAR
DCSDR=DCSDRD/ASTAR
RHOLO=RHOLOD/RHOBAR(1)
ULO=ULOD/ASTAR
LAMDA=LAMDAD*XLCD/ASTAR
MU=MUD*XLCD*PI/ASTAR
XLC=1.0
DO 25 I=1,NVAL
  XBAR(I)=XBARD(I)/XLCD
  UBAR(I)=UBARD(I)/ASTAR
25 CONTINUE
S=CMPLX(LAMDA,MU)
RFAR=(GAMMA-1.0)*UBAR(1)/(2.0*GAMMA)
RFA=CMPLX(RFAR,0.0)
RFC=CMPLX(0.0,0.0)
WRITE(*,*)' '
WRITE(*,*)TITLE
WRITE(*, '( /A, I2/ )') ' Engine No. ', K
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*, '( ' NVAL=' ', I5) ') NVAL
WRITE(*, '( ' XBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (XBARD(I), I=1, NVAL)
WRITE(*, '( ' UBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (UBARD(I), I=1, NVAL)
WRITE(*, 1) VARD(3), HOLDD(3), VARD(4), HOLDD(4), (VARD(I), HOLDD(I),
* I=7, 20)
WRITE(14, 2)
WRITE(14, '(1X, A)') TITLE
WRITE(14, '( /A, I2/ )') ' Engine No. ', K
WRITE(14,*)' '
WRITE(14, '( ' NVAL=' ', I5) ') NVAL
WRITE(14, '( ' XBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (XBARD(I), I=1, NVAL)
WRITE(14, '( ' UBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (UBARD(I), I=1, NVAL)
WRITE(14, 1) VARD(3), HOLDD(3), VARD(4), HOLDD(4), (VARD(I), HOLDD(I),
* I=7, 20)
WRITE(*,*)' '
WRITE(*, '( ' NVAL=' ', I5) ') NVAL
WRITE(*, '( ' XBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (XBARD(I), I=1, NVAL)
WRITE(*, '( ' UBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (UBARD(I), I=1, NVAL)
WRITE(*, 1) (VAR(I), RVAR(I), I=3, 13)
WRITE(*, '( ' RFA=' ', 1P2E13.5, 5X, ' ' RFC=' ', 2E13.5) ') RFA, RFC
WRITE(14, 2)
WRITE(14,*)' '
WRITE(14, '( ' NVAL=' ', I5) ') NVAL
WRITE(14, '( ' XBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (XBARD(I), I=1, NVAL)
WRITE(14, '( ' UBAR=' ', 1P4E13.5/(8X, 4E13.5) ') (UBARD(I), I=1, NVAL)
WRITE(14, 1) (VAR(I), RVAR(I), I=3, 13)
WRITE(14, '( ' RFA=' ', 1P2E13.5, 5X, ' ' RFC=' ', 2E13.5) ') RFA, RFC
WRITE(*, '(A\)'') ' Hit ENTER to continue '
READ(*,*)

```

```

RETURN
END
SUBROUTINE PLTALL(X,Y,NOT,NOF,N,M,LABLX,LABLY,FREQ,NENG)
C   Plots n vs  $\tau$  for all frequencies
DIMENSION X(NOT),Y(NOT,NOF),FREQ(NOF)
CHARACTER*8 LABLX,LABLY,LABFAX(8),LABFAY(8)
CHARACTER*8 XLABL(2),YLABL(2)
CHARACTER*16 FREQL
COMMON /TTTL/TITLE,TITLF,IHR,IMIN,AP,IYR,IMON,IDAY
INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
CHARACTER*2 AP
CHARACTER*13 ETITL
CHARACTER*60 TITLE
CHARACTER*40 TITLF
COMMON /SFACT/SFAC
CHARACTER*8 RADHER(2)
DATA RADHER/' rad/sec',' Hertz '/
DATA LABFAY/' ',' x 10 ',' x 100 ',' x 1000 ','
* ' x 10000',' x-10 ',' x-100 ',' x-1000 '/
DATA LABFAX/' ',' x 10 ',' x 100 ',' x 1000 ','
* ' x 10000',' x-10 ',' x-100 ',' x-1000 '/
DATA ASPECT/1.35/
1 FORMAT(F8.1,A)
2 FORMAT('Engine No. ',I2)
WRITE(ETITL,2)NENG
CALL QRMODE(MODET,NCOLT)
CALL QVIDBD(IBOARD)
IF(IBOARD.LT.1.OR.IBOARD.GT.3) THEN
  WRITE(*,*)' Graphics board not installed!'
  RETURN
ENDIF
IF(IBOARD.EQ.1) MODE=6
IF(IBOARD.EQ.2) MODE=16
IF(IBOARD.EQ.3) MODE=18
YMIN=Y(1,1)
YMAX=Y(N,1)
XMIN=X(1)
XMAX=X(N)
DO 21 I=1,N
  IF(XMIN.GT.X(I)) XMIN=X(I)
  IF(XMAX.LT.X(I)) XMAX=X(I)
DO 21 J=1,M
  IF(YMIN.GT.Y(I,J)) YMIN=Y(I,J)
  IF(YMAX.LT.Y(I,J)) YMAX=Y(I,J)
21 CONTINUE
IF(YMIN.GT.0.0) YMIN=0.0
IXLAB=1
IF(XMAX.LT.1.0) IXLAB=2
IF(XMAX.LT.0.10) IXLAB=3
IF(XMAX.LT.0.010) IXLAB=4
IF(XMAX.LT.0.001) IXLAB=5
IF(XMAX.GT.10.0) IXLAB=6
IF(XMAX.GT.100.0) IXLAB=7

```

```

IF(XMAX.GT.1000.0) IXLAB=8
IYLAB=1
IF(YMAX.LT.1.0) IYLAB=2
IF(YMAX.LT.0.10) IYLAB=3
IF(YMAX.LT.0.010) IYLAB=4
IF(YMAX.LT.0.001) IYLAB=5
IF(YMAX.GT.10.0) IYLAB=6
IF(YMAX.GT.100.0) IYLAB=7
IF(YMAX.GT.1000.0) IYLAB=8
IF(IXLAB.NE.1) THEN
  IF(IXLAB.EQ.2) XFAC=10.0
  IF(IXLAB.EQ.3) XFAC=100.0
  IF(IXLAB.EQ.4) XFAC=1000.0
  IF(IXLAB.EQ.5) XFAC=10000.0
  IF(IXLAB.EQ.6) XFAC=0.1
  IF(IXLAB.EQ.7) XFAC=0.01
  IF(IXLAB.EQ.8) XFAC=0.001
  XMIN=XMIN*XFAC
  XMAX=XMAX*XFAC
  DO 22 I=1,N
    X(I)=X(I)*XFAC
22 CONTINUE
ENDIF
IF(IYLAB.NE.1) THEN
  IF(IYLAB.EQ.2) YFAC=10.0
  IF(IYLAB.EQ.3) YFAC=100.0
  IF(IYLAB.EQ.4) YFAC=1000.0
  IF(IYLAB.EQ.5) YFAC=10000.0
  IF(IYLAB.EQ.6) YFAC=0.1
  IF(IYLAB.EQ.7) YFAC=0.01
  IF(IYLAB.EQ.8) YFAC=0.001
  YMIN=YMIN*YFAC
  YMAX=YMAX*YFAC
  DO 23 J=1,M
    DO 23 I=1,N
      Y(I,J)=Y(I,J)*YFAC
23 CONTINUE
ENDIF
XLABL(1)=LABLX
XLABL(2)=LABFAX(IXLAB)
YLABL(1)=LABLY
YLABL(2)=LABFAY(IYLAB)
XMAJ=0.2*(XMAX-XMIN)
YMAJ=0.2*(YMAX-YMIN)
ICOLR=4
IFIL=3
ILIN=1
CALL QSMODE(MODE)
IF(IBOARD.NE.1) THEN
  CALL QPREG(0,ICOLR)
ENDIF
JCOL1=150
JCOL2=500

```

```

        DO 27 J=1,M
        DO 27 I=1,N
            Y(I,J)=Y(I,J)/YFAC
27 CONTINUE
    ENDIF
    RETURN
    END
    SUBROUTINE PLTVAR(X,Y,N,LABLX,LABLY,FREQ,NENG)
C      Plots n vs  $\tau$  for a single frequency
    DIMENSION X(N),Y(N)
    CHARACTER*8 LABLX,LABLY,LABFAX(8),LABFAY(8)
    CHARACTER*8 XLABL(2),YLABL(2)
    COMMON /TITL/TITLE,TITLF,IHR,IMIN,AP,IYR,IMON,IDAY
    COMMON /SFAC/SFAC
    INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
    CHARACTER*2 AP
    CHARACTER*60 TITLE
    CHARACTER*40 TITLF
    CHARACTER*47 FREQL
    CHARACTER*8 RADHER(2)
    DATA RADHER/' rad/sec',' Hertz '/
    DATA LABFAY/' ',' x 10 ',' x 100 ',' x 1000 ',
*      ' x 10000',' x-10 ',' x-100 ',' x-1000 '/
    DATA LABFAX/' ',' x 10 ',' x 100 ',' x 1000 ',
*      ' x 10000',' x-10 ',' x-100 ',' x-1000 '/
    DATA ASPECT/1.35/
1  FORMAT('Engine No. ',I2,5X,'frequency =',F10.3,A)
    CALL QRMODE(MODET,NCOLT)
    CALL QVIDBD(IBOARD)
    IF(IBOARD.LT.1.OR.IBOARD.GT.3) THEN
        WRITE(*,*)' Graphics board not installed!'
        RETURN
    ENDIF
    IF(IBOARD.EQ.1) MODE=6
    IF(IBOARD.EQ.2) MODE=16
    IF(IBOARD.EQ.3) MODE=18
    XMIN=X(1)
    XMAX=X(N)
    YMIN=Y(1)
    YMAX=Y(N)
    DO 21 I=1,N
        IF(XMIN.GT.X(I)) XMIN=X(I)
        IF(XMAX.LT.X(I)) XMAX=X(I)
        IF(YMIN.GT.Y(I)) YMIN=Y(I)
        IF(YMAX.LT.Y(I)) YMAX=Y(I)
21 CONTINUE
    IF(YMIN.GT.0.0) YMIN=0.0
    IXLAB=1
    IF(XMAX.LT.1.0) IXLAB=2
    IF(XMAX.LT.0.10) IXLAB=3
    IF(XMAX.LT.0.010) IXLAB=4
    IF(XMAX.LT.0.001) IXLAB=5
    IF(XMAX.GT.10.0) IXLAB=6

```



```

      IF(XMAX.GT.100.0) IXLAB=7
      IF(XMAX.GT.1000.0) IXLAB=8
      IYLAB=1
      IF(YMAX.LT.1.0) IYLAB=2
      IF(YMAX.LT.0.10) IYLAB=3
      IF(YMAX.LT.0.010) IYLAB=4
      IF(YMAX.LT.0.001) IYLAB=5
      IF(YMAX.GT.10.0) IYLAB=6
      IF(YMAX.GT.100.0) IYLAB=7
      IF(YMAX.GT.1000.0) IYLAB=8
      IF(IXLAB.NE.1) THEN
        IF(IXLAB.EQ.2) XFAC=10.0
        IF(IXLAB.EQ.3) XFAC=100.0
        IF(IXLAB.EQ.4) XFAC=1000.0
        IF(IXLAB.EQ.5) XFAC=10000.0
        IF(IXLAB.EQ.6) XFAC=0.1
        IF(IXLAB.EQ.7) XFAC=0.01
        IF(IXLAB.EQ.8) XFAC=0.001
        XMIN=XMIN*XFAC
        XMAX=XMAX*XFAC
        DO 22 I=1,N
          X(I)=X(I)*XFAC
22      CONTINUE
      ENDIF
      IF(IYLAB.NE.1) THEN
        IF(IYLAB.EQ.2) YFAC=10.0
        IF(IYLAB.EQ.3) YFAC=100.0
        IF(IYLAB.EQ.4) YFAC=1000.0
        IF(IYLAB.EQ.5) YFAC=10000.0
        IF(IYLAB.EQ.6) YFAC=0.1
        IF(IYLAB.EQ.7) YFAC=0.01
        IF(IYLAB.EQ.8) YFAC=0.001
        YMIN=YMIN*YFAC
        YMAX=YMAX*YFAC
        DO 23 I=1,N
          Y(I)=Y(I)*YFAC
23      CONTINUE
      ENDIF
      XLABL(1)=LABLX
      XLABL(2)=LABFAX(IXLAB)
      YLABL(1)=LABLY
      YLABL(2)=LABFAY(IYLAB)
      XMAJ=0.2*(XMAX-XMIN)
      YMAJ=0.2*(YMAX-YMIN)
      ICOLR=4
      IFIL=3
      ILIN=1
      CALL QSMODE(MODE)
      IF(IBOARD.NE.1) THEN
        CALL QPREG(0,ICOLR)
      ENDIF
      JCOL1=150
      JCOL2=500

```

```

JROW1=40
IF (MODE.EQ.6) JROW1=60
JROW2=149
IF (MODE.EQ.16) JROW2=299
IF (MODE.EQ.18) JROW2=419
XORG=XMIN
YORG=YMIN
YOVERX=1.0
IOPT=0
IF (SFAC.EQ.1.0) THEN
  WRITE (FREQ,1) NENG, FREQ, RADHER(1)
ELSE
  WRITE (FREQ,1) NENG, FREQ, RADHER(2)
ENDIF
IF (MODE.NE.18) THEN
  CALL QPTXT(60,TITLE,7,5,23)
  CALL QPTXT(47,FREQ,7,25,22)
ELSE
  CALL QPTXT(60,TITLE,7,5,29)
  CALL QPTXT(47,FREQ,7,25,28)
ENDIF
CALL QPTXT(8,YLABL(1),7,2,15)
CALL QPTXT(8,YLABL(2),7,2,14)
CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
*          XORG,YORG,IOPT,YOVERX,ASPECT)
CALL QSETUP(0,ILIN,-2,IFIL)
CALL QXAXIS(XMIN,XMAX,0.0,0,0,0)
CALL QPTXTA(16,XLABL,7)
CALL QXAXIS(XMIN,XMAX,XMAJ,0,-1,2)
CALL QYAXIS(YMIN,YMAX,YMAJ,0,-1,2)
CALL QTABL(1,N,X,Y)
24 CONTINUE
CALL QONKEY(IKEY)
IF (IKEY.EQ.0) GO TO 24
CALL QINKEY(IXTEN,IKEY)
CALL QSMODE(MODET)
25 CONTINUE
IF (IXLAB.NE.1) THEN
  DO 26 I=1,N
    X(I)=X(I)/XFAC
26 CONTINUE
  ENDOF
IF (IYLAB.NE.1) THEN
  DO 27 I=1,N
    Y(I)=Y(I)/YFAC
27 CONTINUE
  ENDOF
RETURN
END
SUBROUTINE READIN
C Reads input data
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*          S,GF,GOX,RFA,RFC

```

```

COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*      DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
COMMON /DIMVAL/HOLDD(20),XBARD(50),PBAR(50),TBAR(50)
COMMON /TITL/TITLE,TITLF,IHR,IMIN,AP,IYR,IMON,IDAY
INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
CHARACTER*2 AP
CHARACTER*60 TITLE
CHARACTER*40 TITLF
REAL MBAR,N,NR,LAMDA,MU,RVAR(15)
REAL MBARD,ND,NRD,LAMDAD,MUD,HOLD(20)
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX CVAR(17)
EQUIVALENCE (N,RVAR(1)),(X1,CVAR(1))
EQUIVALENCE (ND,HOLD(1)),(TAUD,HOLD(2)),(DTAUD,HOLD(3)),
*          (NRD,HOLD(4)),(LAMDAD,HOLD(5)),(MUD,HOLD(6)),
*          (CDIAM,HOLD(7)),(TDIAM,HOLD(8)),(XLCD,HOLD(9)),
*          (GAMMAD,HOLD(10)),(RGAS,HOLD(11)),(P00D,HOLD(12)),
*          (MBARD,HOLD(13)),(RBARD,HOLD(14)),(DCSDRD,HOLD(15)),
*          (DHLDRD,HOLD(16)),(RHOLOD,HOLD(17)),(ULOD,HOLD(18)),
*          (PCHMB,HOLD(19)),(TCHMB,HOLD(20))
CHARACTER*8 VAR(20),VARP(20),VARL(20),NAME
CHARACTER*1 ANS
COMMON /EPARAM/MENG,TFLOW(25),PCHMBX(25),DPROR(25),PMRAT(25)
COMMON /WORK/YP1(20,25),YP2(50,25),YP3(50,25),YP4(50,25),
*          YP5(30,25)
DATA IGO/0/
DATA VAR /'      ND =' , '      TAUD =' , '      DTAUD =' , '      NRD =' , 'LAMDAD =' ,
*          '      MUD =' , '      CDIAM =' , '      TDIAM =' , '      XLCD =' , 'GAMMAD =' ,
*          '      RGAS =' , '      P00D =' , '      MBARD =' , '      RBARD =' , 'DCSDRD =' ,
*          'DHLDRD =' , 'RHOLOD =' , '      ULOD =' , '      PCHMB =' , '      TCHMB =' /
DATA VARP/'ND      ' , 'TAUD      ' , 'DTAUD      ' , 'NRD      ' , 'LAMDAD      ' ,
*          'MUD      ' , 'CDIAM      ' , 'TDIAM      ' , 'XLCD      ' , 'GAMMAD      ' ,
*          'RGAS      ' , 'P00D      ' , 'MBARD      ' , 'RBARD      ' , 'DCSDRD      ' ,
*          'DHLDRD      ' , 'RHOLOD      ' , 'ULOD      ' , 'PCHMB      ' , 'TCHMB      ' /
DATA VARL/'nd      ' , 'taud      ' , 'dtaud      ' , 'nrd      ' , 'lamdad      ' ,
*          'mud      ' , 'cdiam      ' , 'tdiam      ' , 'xlcd      ' , 'gammad      ' ,
*          'rgas      ' , 'p00d      ' , 'mbard      ' , 'rbard      ' , 'dcsdrd      ' ,
*          'dhldrd      ' , 'rholod      ' , 'ulod      ' , 'pchmb      ' , 'tchmb      ' /
1 FORMAT(16I5)
2 FORMAT(' Enter X (ft), P (lbf/ft^2), and T (°R) for point ',
*          I3,' ')
3 FORMAT(1P4E15.6)
4 FORMAT(2X,A8,1PE13.5,2X,A8,E13.5,2X,A8,E13.5)
5 FORMAT(1P3E15.6)
6 FORMAT(A40,2X,I2.2,':',I2.2,A2,3X,I2.2,'-',I2.2,'-',I2.2)
  IF(IGO.EQ.1) THEN
    WRITE(*,'(A\)' )' Do you wish to use old data with or without chan
*ges? Y or N '
    READ(*,'(A)' )ANS
    IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') GO TO 24
  ENDIF
  IGO=1

```

```

IDATA=1
WRITE(*,*) ' '
WRITE(*, '(A\)' ) ' Is your rocket input on file? Y OR N '
READ(*, '(A)' )ANS
IF(ANS.NE.'N'.AND.ANS.NE.'n') THEN
  WRITE(*, '(A\)' ) ' Does the file need to be rewound? Y OR N '
  READ(*, '(A)' )ANS
  IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') REWIND 12
  READ(12, '(A)', END=99) TTTLF
  WRITE(TITLE, 6) TTTLF, IHR, IMIN, AP, IMON, IDAY, IYR
  IDATA=0
ELSE
  WRITE(*,*) ' Enter Title'
  READ(*, '(A)' ) TTTLF
  WRITE(TITLE, 6) TTTLF, IHR, IMIN, AP, IMON, IDAY, IYR
  WRITE(12, '(A)' ) TTTLF
ENDIF
REWIND 20
DO 23 K=1, MENG
  IF(IDATA.EQ.0) THEN
    READ(12, *, END=99) NVAL
    IF(NVAL.EQ.0) GO TO 99
    DO 21 I=1, NVAL
      READ(12, *) XBARD(I), PBAR(I), TBAR(I)
21    CONTINUE
      READ(12, *) DTAUD, NRD, LAMDA
      READ(12, *) CDIAM, TDIAM, XLCD
      READ(12, *) GAMMAD, RGAS, P00D, RBARD
      READ(12, *) DCSDRD, DHLDRD, RHOLOD, ULOD
      MBARD=TFLOW(K)
    ELSE
      WRITE(*, '(A\)' ) ' How many points along centerline? '
      READ(*, *, END=99) NVAL
      IF(NVAL.EQ.0) GO TO 99
      DO 22 I=1, NVAL
        WRITE(*, 1) I
        READ(*, *) XBARD(I), PBAR(I), TBAR(I)
22    CONTINUE
        WRITE(*,*) ' Enter NR (mixture ratio interaction index) '
        READ(*, *) NR
        WRITE(*,*) ' Enter DTAU (invariant time lag - sec) '
        READ(*, *) DTAUD
        WRITE(*,*) ' Enter LAMDA (real part of frequency) '
        READ(*, *) LAMDA
        WRITE(*,*) ' Enter XLCD (length of combustion chamber - ft) '
        READ(*, *) XLCD
        WRITE(*,*) ' Enter CDIAM (chamber diameter - ft) and TDIAM',
*         ' (throat diameter - ft) '
        READ(*, *) CDIAM, TDIAM
        WRITE(*,*) ' Enter GAMMA (ratio of specific heats), RGAS',
*         ' (gas constant - ft^2/sec^2/°R) '
        READ(*, *) GAMMAD, RGAS
        WRITE(*,*) ' Enter P00 (maximum overpressure - lbf/ft^2) '

```

```

      READ(*,*)POOD
      WRITE(*,*)' Enter RBAR (mean mixture ratio)'
      READ(*,*)RBARD
      WRITE(*,*)' Enter DCSDR (dc*/dr - ft/sec) and DHLDR',
*      ' (dh/dr - ft^2/sec^2)'
      READ(*,*)DCSDRD,DHLDRD
      WRITE(*,*)' Enter RHOLO (mass of liquid/unit chamber vol -',
*      ' lbm/ft^3)'
      WRITE(*,*)' and ULO (axial component of liquid velocity',
*      ' - ft/sec)'
      READ(*,*)RHOLOD,ULOD
      MBARD=TFLOW(K)
      WRITE(12,1)NVAL
      WRITE(12,5)(XBARD(I),PBAR(I),TBAR(I),I=1,NVAL)
      WRITE(12,3)DTAUD,NR,LAMDAD
      WRITE(12,3)CDIAM,TDIAM,XLCD
      WRITE(12,3)GAMMAD,RGAS,POOD,RBARD
      WRITE(12,3)DCSDRD,DHLDRD,RHOLOD,ULOD
      ENDIF
      PCHMB=PCHMBX(K)
      TCHMB=TBAR(1)
      CALL NONDIM(HOLD,K)
      WRITE(20)HOLDD,XBARD,PBAR,TBAR
23  CONTINUE
      RETURN
24  CONTINUE
      WRITE(*, '(A\)' )' are there any changes? Y or N '
      READ(*, '(A)' )ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'y') RETURN
      WRITE(*, '(A\)' )' Do you wish to change title? Y or N '
      READ(*, '(A)' )ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
        WRITE(*,*)' Enter Title'
        READ(*, '(A)' )TTTLF
        WRITE(TITLE,6)TTTLF,IHR,IMIN,AP,IMON,IDAY,IYR
      ENDIF
      REWIND 20
      DO 33 K=1,MENG
        READ(20)HOLDD,XBARD,PBAR,TBAR
        WRITE(*, '(A,I2,A\)' )' are there any changes for engine no. ',
*      K, ' ? '
        READ(*, '(A)' )ANS
        IF(ANS.NE.'Y'.AND.ANS.NE.'y') GO TO 31
        GO TO 27
25  CONTINUE
      WRITE(*,*)' VARIABLE NAMES AND DESCRIPTIONS'
      WRITE(*,*)' '
      WRITE(*,*)' DTAUD - invariant time lag sec'
      WRITE(*,*)' NRD - mixture ratio interaction index'
      WRITE(*,*)' LAMDAD - damping of perturbation'
      WRITE(*,*)' CDIAM - chamber diameter ft'
      WRITE(*,*)' TDIAM - throat diameter ft'
      WRITE(*,*)' XLCD - length of combustion chamber ft'

```

```

WRITE(*,*) ' GAMMAD - ratio of specific heats'
WRITE(*,*) ' RGAS   - gas constant',
* ' (ft/sec)^2/°R'
WRITE(*,*) ' P00D   - maximum pressure',
* ' lbf/ft^2'
WRITE(*,*) ' RBARD   - mean mixture ratio'
WRITE(*,*) ' DCSDRD  - d(c*)/d(mixture ratio)',
* ' ft/sec'
WRITE(*,*) ' DHLDRD  - d(enthalpy)/d(mixture ratio)',
* ' ft^2/sec^2'
WRITE(*,*) ' RHOLOD  - mass of liquid/unit chamber volume',
* ' lbm/ft^3'
WRITE(*,*) ' ULOD    - axial component of liquid velocity',
* ' ft/sec'
WRITE(*,*) ' TCHMB   - chamber temperature °R'
WRITE(*,*) ' '
GO TO 28
26 CONTINUE
WRITE(*,*) ' VARIABLE NAMES AND VALUES'
WRITE(*,*) ' '
WRITE(*,4) (VAR(I),HOLD(I),I=3,5), (VAR(I),HOLD(I),I=7,12),
* (VAR(I),HOLD(I),I=14,18), VAR(20),HOLD(20)
27 CONTINUE
WRITE(*,*) ' '
WRITE(*,*) ' Enter ? to print variable names & descriptions'
WRITE(*,*) ' # to print variable names & values'
WRITE(*,*) ' END when all changes have been made'
WRITE(*,*) ' '
28 CONTINUE
WRITE(*, '(A\)' ) ' Enter variable name and new value, END, ?, or
* # '
CALL ZREAD(NAME,VALUE)
IF(NAME.EQ.'?') GO TO 25
IF(NAME.EQ.'#') GO TO 26
IF(NAME.EQ.'END'.OR.NAME.EQ.'end') THEN
CALL NONDIM(HOLD,K)
GO TO 31
RETURN
ENDIF
DO 29 II=3,20
IF(II.EQ.6.OR.II.EQ.13.OR.II.EQ.19) GO TO 29
I=II
IF(NAME.EQ.VARP(I).OR.NAME.EQ.VARL(I)) GO TO 30
29 CONTINUE
WRITE(*,*) ' Invalid name, try again'
GO TO 25
30 CONTINUE
HOLD(I)=VALUE
GO TO 28
31 CONTINUE
DO 32 J=1,50
IF(J.LE.20) YP1(J,K)=HOLDD(J)
YP2(J,K)=XBARD(J)

```

```

    AVGK(MM)=0.0
    DIVAVG=0.0
    READ(IUNIT,*) SEGMN(M), SPLIT(MM)
    DO 21 I=1,SEGMN(M)
        READ(IUNIT,*) SECTN(I,M), PIPE1(I,M), PIPE2(I,M), PIPE3(I,M),
*           PIPE4(I,M), PIPE5(I,M)
        IF(SECTN(I,M).NE.7) GO TO 21
        AVGK(MM)=AVGK(MM)+PIPE2(I,M)
        DIVAVG=DIVAVG+1
21 CONTINUE
    IF(SPLIT(MM).EQ.0) THEN
        AVGK(MM)=KTANK(IT)
        GO TO 24
    ENDIF
C       split pipe
    DO 23 J=1,SPLIT(MM)
        M=M+1
        READ(IUNIT,*) SEGMN(M), NOLINE(M), IENG(M)
        IF(IENG(M).GT.MENG) THEN
            WRITE(*,*) ' Invalid engine number.'
            STOP
        ENDIF
        IE=IENG(M)
        IF(NOLINE(M).EQ.0) NOLINE(M)=1
        DO 22 I=1,SEGMN(M)
            READ(IUNIT,*) SECTN(I,M), PIPE1(I,M), PIPE2(I,M), PIPE3(I,M),
*           PIPE4(I,M), PIPE5(I,M)
            IF(SECTN(I,M).NE.7) GO TO 22
            AVGK(MM)=AVGK(MM)+PIPE2(I,M)*NOLINE(M)
            DIVAVG=DIVAVG+NOLINE(M)
22 CONTINUE
23 CONTINUE
        WRITE(*, '(A,I3)') ' Max. no. of iterations is set at ',
*           LOPOLD(MM)
        WRITE(*, '(A\)\') ' Do you wish to change it? '
        READ(*, '(A)\') ANS
        IF(ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
            WRITE(*, '(A\)\') ' Enter maximum no. of iterations '
            READ(*,*) LOPOLD(MM)
        ENDIF
        LOPEND(MM)=LOPOLD(MM)
        IF(DIVAVG.LE.0.0) DIVAVG=1.0
        AVGK(MM)=KTANK(IT)+AVGK(MM)/DIVAVG
24 CONTINUE
    M=0
    DO 28 MM=1,MLINE
        M=M+1
        IT=ITANK(MM)
        IE=IENG(M)
        DO 25 I=1,SEGMN(M)
            CALL RTYPE(SECTN(I,M), PIPE1(I,M), PIPE2(I,M),
*           PIPE3(I,M), PIPE4(I,M), PIPE5(I,M), L(I,M), AREA(I,M),
*           DIA(I,M), PIND(I,M), PCAP(I,M), AVGK(MM), DENS(IT),

```

```

*          CMAN(M) , KMAN(M) , VOLMF(M) )
25 CONTINUE
  IF(SPLIT(MM).EQ.0) GO TO 28
  DO 27 J=1,SPLIT(MM)
    M=M+1
    IE=IENG(M)
    DO 26 I=1,SEGMN(M)
      CALL RTYPE(SECTN(I,M) , PIPE1(I,M) , PIPE2(I,M) ,
*              PIPE3(I,M) , PIPE4(I,M) , PIPE5(I,M) , L(I,M) , AREA(I,M) ,
*              DIA(I,M) , PIND(I,M) , PCAP(I,M) , AVGK(MM) , DENS(IT) ,
*              CMAN(M) , KMAN(M) , VOLMF(M) )
26 CONTINUE
27 CONTINUE
28 CONTINUE
  RETURN
  END
  SUBROUTINE RTYPE(SECTN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,L,
*                AREA,DIA,PIND,PCAP,AVGK,DENS,CMAN,KMAN,
*                VOLMF)
C      Stores values for different types of piping
  INTEGER SECTN
  REAL L,KMAN
  DATA GRAV/32.2/,PI/3.141593/
  IF(SECTN.EQ.0) THEN
    CALL BENDS(PIPE1,PIPE2,PIPE3,PIPE4,VALUE,DIME)
    AREAB=0.785398*DIME**2
    L=VALUE
    AREA=AREAB
    DIA=DIME
  ELSEIF(SECTN.EQ.1) THEN
C      straight section
    VALUE=PIPE1
    DIME=PIPE2
    AREAB=0.785398*DIME**2
    L=VALUE
    AREA=AREAB
    DIA=DIME
  ELSEIF(SECTN.EQ.2) THEN
C      inline accumulator
C      PIPE1 - LEN
C      PIPE2 - DIA
C      PIPE3 - DEN
C      PIPE4 - K
    L=PIPE1
    DIA=PIPE2
    AREA=0.25*PI*PIPE2**2
    IF(PIPE3.EQ.0.0) PIPE3=DENS
    IF(PIPE4.EQ.0.0) PIPE4=AVGK
    PCAP=PIPE3*L*AREA/PIPE4
  ELSEIF(SECTN.EQ.3) THEN
C      tuned stub - suppresses omega = (PI/2)/(L*SQRT(PIND*PCAP))
C      PIPE1 - LEN
C      PIPE2 - DIA

```



```

C      PIPE3 - DEN
C      PIPE4 - K
      L=PIPE1
      DIA=PIPE2
      AREA=0.25*PI*DIA**2
      IF(PIPE3.EQ.0.0) PIPE3=DENS
      IF(PIPE4.EQ.0.0) PIPE4=AVGK
      PCAP=PIPE3*L*AREA/PIPE4
      PIND=L/(AREA*GRAV)
      ELSEIF(SECTN.EQ.4.OR.SECTN.EQ.5) THEN
C          helmholtz resonator or parallel resonator
C          suppresses omega = 1/SQRT(PIND*PCAP)
C      PIPE1 - LEN
C      PIPE2 - DIA
C      PIPE3 - VOL
C      PIPE4 - DEN
C      PIPE5 - K
      L=PIPE1
      DIA=PIPE2
      AREA=PIPE3
      IF(PIPE4.EQ.0.0) PIPE4=DENS
      IF(PIPE5.EQ.0.0) PIPE5=AVGK
      PCAP=PIPE4*AREA/PIPE5
      PIND=L/(0.25*PI*DIA**2*GRAV)
      ELSEIF(SECTN.EQ.6) THEN
C          pump
C      PIPE1 - LEN
C      PIPE2 - DIA
C      PIPE3 - DP/DM
C      PIPE4 - IND
C      PIPE5 - CAP
      L=PIPE1
      DIA=PIPE2
      AREA=PIPE3
      PCAP=PIPE4
      PIND=PIPE5
      ELSEIF(SECTN.EQ.7) THEN
C          manifold
C      PIPE1 - VOLMF
C      PIPE2 - KMAN
      VOLMF=PIPE1
      KMAN=PIPE2
      CMAN=DENS*VOLMF/KMAN
      L=VOLMF
      DIA=CMAN
      ENDIF
      RETURN
      END
C      SUBROUTINE SETVAL(VAL,ID)
      Sets value from iterated variable
      COMMON /DIMVAL/HOLDD(20),XBARD(50),PBAR(50),TBAR(50)
      VAL=HOLDD(ID)
      RETURN

```

```

END
SUBROUTINE SETVAR(VAL, ID)
C   Sets iterated variable from value
COMMON /CMPVAL/X1,Y1,Z1,W1,M1,P0,P1,U0,U1,RFH,RFK,RFP,
*       S,GF,GOX,RFA,RFC
COMMON /RELVAL/N,TAU,DTAU,NR,RBAR,MBAR,GAMMA,P00,DHLDR,CSTAR,
*       DCSDR,RHOLO,ULO,LAMDA,MU,TAUT,UBAR(50),XBAR(50),XLC
COMMON /INTVAL/NVAL
COMMON /DIMVAL/HOLDD(20),XBARD(50),PBAR(50),TBAR(50)
REAL MBAR,N,NR,LAMDA,MU,RVAR(13)
REAL MBARD,ND,NRD,LAMDAD,MUD
COMPLEX S,X1,Y1,Z1,W1,M1,P0,P1,U0,U1,GF,GOX,RFH,RFK,RFP,RFA,RFC
COMPLEX CVAR(17)
EQUIVALENCE (N,RVAR(1)), (X1,CVAR(1))
EQUIVALENCE
*   (ND,HOLDD(1)), (TAUD,HOLDD(2)), (DTAUD,HOLDD(3)),
*   (NRD,HOLDD(4)), (LAMDAD,HOLDD(5)), (MUD,HOLDD(6)),
*   (CDIAM,HOLDD(7)), (TDIAM,HOLDD(8)), (XLCD,HOLDD(9)),
*   (GAMMAD,HOLDD(10)), (RGAS,HOLDD(11)), (P00D,HOLDD(12)),
*   (MBARD,HOLDD(13)), (RBARD,HOLDD(14)), (DCSDRD,HOLDD(15)),
*   (DHLDRD,HOLDD(16)), (RHOLOD,HOLDD(17)), (ULOD,HOLDD(18)),
*   (PCHMB,HOLDD(19)), (TCHMB,HOLDD(20))
DATA PI/3.141593/
HOLDD(ID)=VAL
IF (ID.EQ.1) THEN
C       ND
        N=ND
        RETURN
    ENDIF
IF (ID.EQ.2) THEN
C       TAUD
        ASTAR=SQRT(GAMMAD*RGAS*TBAR(1))
        TAU=TAUD*ASTAR/XLCD
        TAUT=TAU+DTAU
        RETURN
    ENDIF
IF (ID.EQ.6) THEN
C       MUD
        ASTAR=SQRT(GAMMAD*RGAS*TBAR(1))
        MU=MUD*XLCD*PI/ASTAR
        S=CMPLX(LAMDA,MU)
        RETURN
    ENDIF
RETURN
END
SUBROUTINE TANKNO(MTANK,VOL,LFLOW,KTANK,DENS,A,CTANK,IUNIT)
C   Reads tank parameters
REAL VOL(25),LFLOW(25),KTANK(25),DENS(25),A(25),CTANK(25)
DATA GRAV/32.2/
READ(IUNIT,*)MTANK
IF (MTANK.GT.25) THEN
    WRITE(*,*) 'Number of tanks must be less than 25'
    STOP

```

```

ENDIF
IF(MTANK.LE.0) MTANK=1
DO 21 I=1,MTANK
READ(IUNIT,*)VOL(I),LFLOW(I),KTANK(I),DENS(I)
A(I)=SQRT(GRAV*KTANK(I)/DENS(I))
CTANK(I)=DENS(I)*VOL(I)/KTANK(I)
21 CONTINUE
RETURN
END
SUBROUTINE ZREAD(NAME,VALUE)
C   Reads input for input modification
CHARACTER*1 NAME(8)
CHARACTER*1 CARD(80),PLUS,MINUS,PERIOD,LE,E,NUMBER(10)
CHARACTER*1 LEND(3),CEND(3),POUND,QUEST,BLK,COMMA
CHARACTER*80 DCARD
EQUIVALENCE (CARD(1),DCARD)
DATA PLUS/'+'/,MINUS/'-'/,PERIOD/'.'/,LE/'e'/,E/'E'/,BLK/' '/
DATA NUMBER/'0','1','2','3','4','5','6','7','8','9'/,COMMA/','/,
DATA LEND/'e','n','d'/,CEND/'E','N','D'/,POUND/'#'/,QUEST/'?'/
1 FORMAT(A)
DO 21 I=1,8
NAME(I)=BLK
21 CONTINUE
READ(*,1)DCARD
IF(CARD(1).EQ.POUND) THEN
NAME(1)=POUND
RETURN
ENDIF
IF(CARD(1).EQ.QUEST) THEN
NAME(1)=QUEST
RETURN
ENDIF
DO 22 I=1,3
IF(CARD(I).NE.LEND(I).AND.CARD(I).NE.CEND(I)) GO TO 23
NAME(I)=CEND(I)
22 CONTINUE
RETURN
23 CONTINUE
DO 24 I=1,8
II=I
IF(CARD(I).EQ.BLK.OR.CARD(I).EQ.COMMA) GO TO 25
NAME(I)=CARD(I)
24 CONTINUE
25 CONTINUE
DO 26 I=II,80
ID=I
IF(CARD(I).NE.BLK.AND.CARD(I).NE.COMMA) GO TO 27
26 CONTINUE
VALUE=0.0
WRITE(*,*) ' No value given, ZERO assumed'
RETURN
27 CONTINUE
SIGN=1.0

```

```

      IF(CARD(ID).EQ.MINUS) THEN
        SIGN=-1.0
        ID=ID+1
      ELSEIF(CARD(ID).EQ.PLUS) THEN
        ID=ID+1
      ENDIF
      WHOLE=0.0
      DO 30 I=ID,80
        II=I
        IF(CARD(I).EQ.PERIOD) GO TO 31
        IF(CARD(I).EQ.PLUS) GO TO 36
        IF(CARD(I).EQ.MINUS) GO TO 36
        IF(CARD(I).EQ.E.OR.CARD(I).EQ.LE) GO TO 35
        DO 28 J=1,10
          JJ=J-1
          IF(CARD(I).EQ.NUMBER(J)) GO TO 29
28      CONTINUE
          VALUE=SIGN*WHOLE
          IF(CARD(I).EQ.BLK) RETURN
          WRITE(*,*)' Input error, value set to ZERO'
          VALUE=0.0
          RETURN
29      CONTINUE
          WHOLE=WHOLE*10.0+JJ
30      CONTINUE
          VALUE=SIGN*WHOLE
          RETURN
31      CONTINUE
          ID=II+1
          FRACT=0.0
          ICOUNT=0
          DO 34 I=ID,80
            ICOUNT=ICOUNT+1
            II=I
            IF(CARD(I).EQ.PERIOD) THEN
              WRITE(*,*)' Input error, value set to ZERO'
              VALUE=0.0
              RETURN
            ENDIF
            IF(CARD(I).EQ.PLUS) GO TO 36
            IF(CARD(I).EQ.MINUS) GO TO 36
            IF(CARD(I).EQ.E.OR.CARD(I).EQ.LE) GO TO 35
            DO 32 J=1,10
              JJ=J-1
              IF(CARD(I).EQ.NUMBER(J)) GO TO 33
32      CONTINUE
              VALUE=SIGN*(WHOLE+FRACT)
              IF(CARD(I).EQ.BLK) RETURN
              WRITE(*,*)' Input error, value set to ZERO'
              VALUE=0.0
              RETURN
33      CONTINUE
          FRACT=FRACT+JJ/10.0**ICOUNT

```

```

34 CONTINUE
   VALUE=SIGN*(WHOLE+FRACT)
   RETURN
35 CONTINUE
   II=II+1
36 CONTINUE
   VALUE=SIGN*(WHOLE+FRACT)
   SIGN=1.0
   IF(CARD(II).EQ.MINUS) THEN
     SIGN=-1.0
     II=II+1
   ELSEIF(CARD(II).EQ.PLUS) THEN
     II=II+1
   ENDIF
   WHOLE=0.0
   DO 39 I=II,80
     DO 37 J=1,10
       JJ=J-1
       IF(CARD(I).EQ.NUMBER(J)) GO TO 37
37 CONTINUE
       VALUE=VALUE*10.0**(SIGN*WHOLE)
       IF(CARD(I).EQ.BLK) RETURN
       WRITE(*,*) ' Input error, value set to ZERO'
       VALUE=0.0
       RETURN
38 CONTINUE
       WHOLE=WHOLE*10.0+JJ
39 CONTINUE
       VALUE=VALUE*10.0**(SIGN*WHOLE)
       RETURN
   END

```